Online Automated G&G Data Analysis for Locating Oil & Gas Well

by

Hamizan Binti Sharbini

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Communication & Technology)

JANUARY 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

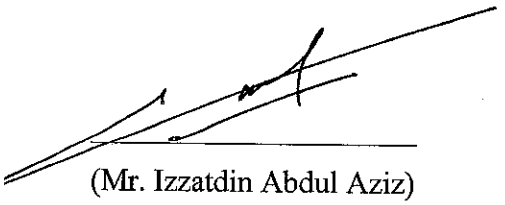
Online Automated G&G Data Analysis for Locating Oil & Gas Well

by

Hamizan Binti Sharbini

A project dissertation submitted to the
Information Communication Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION AND COMMUNICATION TECHNOLOGY)

Approved by,



(Mr. Izzatdin Abdul Aziz)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

JANUARY 2006

CERTICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Hamizan Binti Sharbini

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to the Al-Mighty for all of the blessings that I can uphold during the progress of doing my final year project. I wish to also express my appreciation to my dedicated FYP supervisor, Mr. Izzatdin Abdul Aziz for commission this project with so much effort and giving guidelines during these two semesters. Not to be forgotten, this appreciation is also dedicated to the head of Geo-Chemical Department from Petronas Research Scientific & Services (PRSS), Mr. Abdul Jalil Muhamad who has been the foremost person in contributing partial information of well data which is related to this project. Furthermore, I would like to thank my fellow colleagues who indirectly involved in the entire process of conceptualizing and reviewing during the development of the system and to my beloved parents who had supported me both financially and spiritually. Finally, I thank Universiti Teknologi PETRONAS for providing the facilities which is necessary as part of this final year project achievement.

ABSTRACT

This dissertation mainly focusing on a system that analyses data to overcome the problem of Geochemical and Geosciences (G&G) data analysis, which is time consuming and yet, the results of data analysis are less accurate. The objective of this study is to develop an online system that can give conclusion and generate a chart based on data analysis for locating well automatically. The scope of study will cover on the area of G&G data, generating particular charts and to give prediction or interpretation of the well location and how to develop the system. The methodology for the system is using evolutionary development which caters on user requirement elements. The discussion and results section will discuss on the objectives that has been met, user of the system and the system flow. The system flow includes the explanation of each functional system's module. As a conclusion, a near to accurate and quick G&G chart data analysis is essential to overcome the limitation of using spreadsheet program. Data interpretation is presented according to the selected value from the generate chart process that will determine the range for immature, oil window or gas window.

TABLE OF CONTENTS

ABBREVIATIONS

ASP

Active Server Page

G&G

Geological & Geosciences

GIS

Geographical Information System

IIS

Internet Information Server

PRSS

Petronas Research Scientific & Services

TOC

Total Organic Carbon

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. OVERVIEW

Existing G&G data analysis system is highly expensive. Due to that reason, it is essential to develop an efficient system with cheaper means. The system that is yet to be developed should incorporate user friendliness and less complexity. The capability of doing data interpretation for better judgment of potential formation of source rock and handling massive data accurately will lead to precise prediction of well location.

1.1 Background of Study

It is crucial to understand what G&G data before embarking on the analysis part. Therefore, study on geo-chemical data is vital for betterment of understanding. To gain better outcome of the project, research elements such as methodology, fact-findings and others have been taken into account while conducting the study.

1.1.1 General G&G Data Analysis

The detection of data outliers and unusual data structures is one of the main tasks in the statistical analysis of geo-chemical data. Traditionally, despite the fact that geochemistry data sets are almost always multivariate, outliers are most frequently sought for each single variable in a given data set [Reimann et al., 2005].

This shows that a good organization of data set is important when dealing with geo-chemical data to achieve accuracy in results.

Knowledge of statistical methods for software engineering is becoming increasing important due to industry trends [Card, 2004] as well as the increasing rigor adopted in empirical research. The geochemistry data can be massive and thus statistical analysis is crucial to assist engineering professionals such as measurement analyst, quality assurance personnel, process improvement specialists, technical leads and managers.

From this, we can rationalize that a user friendly and less complex system would be helpful in order to assist these technical people when dealing with massive data. Statistical analysis is becoming an increasingly important skill for software engineering practitioners and researchers. These techniques involve the rigorous collection of data, development of statistical models describing that data, and application of those models to decision-making by managers and engineers. The result is better decisions with a known level of confidence.

In recent years, chemical analytical data in sedimentary petrology have been accumulated and used to estimate the sedimentary process and/or province tectonic setting together with mathematical analysis of data [Bhatia, 1983; Fedo *et al.*, 1995; Kiminami *et al.*, 1998]. Quantitative research [Lacey; Luff, 2001] techniques generate a mass of numbers that need to be summarized, described and analyzed.

Characteristics of the data may be described and explored by drawing graphs and charts, doing cross tabulations and calculating means and standard deviations. Further analysis would build on these initial findings, seeking patterns and relationships in the data by performing multiple regressions, or an analysis of variance perhaps.

This can be achieved by plotting geo-chemical data to generate graph. However, the accuracy of the graph may still be questionable due to random sampling for plotting.

1.2 Problem Statement

1.2.1 Problem Identification

G&G data consists of massive geo-chemical data; there is no specific database to store all the data. The Geo-Chemical Department at PRSS have been using spreadsheets like Lotus 123 and Microsoft Excel to store all the data, which can be up to thousands data per sample source (rock, gas or soil). This however contributes to disorganize data management.

The problem comes into surface when geochemist technician have to plot the geo-chemical data in a chart form. Processing massive data leads to time-consuming effort and inaccuracy of manual plotting due to random sampling

There's no online system developed for G&G data for graph plotting in Geo-Chemical Department at PRSS available yet. The summary log derived from manually plotted charts needs to be included in report, which is in paper-based form. Other departments in PRSS that collaborate with Geo-chemical Department had difficulty in retrieving the information as for now they still have to go directly to Geo-chemical Department to get the information required hand to hand.

1.2.2 Significance of the Project

The system will be useful for Geo-chemical Department at PRSS in order to organize massive geo-chemical data and effectively produce an accurate geo-chemical plotting point. Through online or web-base system, the data can be accessed by all from

various locations with authentication to further progress in their work related to data sample taken. A near to accurate prediction of the well's location will eventually reduce the time for the exploration.

1.3 Objectives and Scopes of Study

1.3.1 Objectives

The objectives of developing Online and Automated G&G Data Analysis for Locating Oil and Gas Well system are:

- To study on how data analysis is perform when dealing with massive geo-chemical data.
    - The study is conducted through an interview with the head of Geoscience's Group from PRSS, which is located in Bangi. An open-ended questionnaire had been conducted during the interview session. Fact-finding including internet research and journals or books.

- To develop a system capable to perform data analysis and charting.
    - The system will provide fast and near to accurate plotting of geo-chemical data. Thus, the system is expected to have the ability to generate charts automatically.
    - To make an analysis upon geo-chemical data plots will help determine which well has lots of good maturity of source rocks that can be further explored.
    - The clients can easily view the geo-chemical summary log from the online system developed.

1.3.2 Scopes of Study

The scopes of study are mainly focus on G&G data, charting of data to give near to accurate prediction or interpretation of the well location and how to develop the system.

G & G Data

The Geo-chemical Research performs high quality analysis in inorganic and organic geochemistry. Petroleum system studies the genetic relationship between pods of active source rock describe and the resulting oil and gas accumulations. A source rock is the sedimentary rocks that are, or may become, or have been able to generate petroleum. A source rock that is very effective can generate expelled petroleum. Thus, an immature source rock, which consists of geo-chemical parameters, is describing potential petroleum availability. (See Appendix Section A for Figure 1.3.2.1: Parameters of Immature Source Rock)

As the parameters from the sample rock attribute can be very massive, it is necessary to have all the geo-chemical data to be stored in repository. The geo-chemical data such as TOC and Hydrocarbon parameters are generated in table form from a machine resides in geo-chemical laboratory. The table represents the geo-chemical data from a source rock sample that will need to be analyzed later on.

The data analyzing part consist of plotting an oil well-detail-information chart. The data analysis is done prior to its oil well locating basis. In other words, the geo-chemical data from a sample source rocks taken will be plotted into chart. From the plotted geo-chemical data, the chart can give analysis in terms of tabulated data of the sample rocks. A good geo-chemical data analysis can give good petroleum potential that deserves better oil well exploration or oil well drilling in particular area. Thus, it will involve study regarding G&G data as well as data analysis pertaining to locating oil and gas well.

Data analysis is essential for determining the oil well location that has good sources of rocks sample. Thus, a near to accurate data plotting is crucial. Category of charts that is suitable to do the plotting will be taken into consideration as well.

Types of Charts

Data that have been collected need to be reviewed carefully. By using charts and graphs, it will help in organizing data and patterns that have been collected. In other words, the needs to summarize data should be taken into consideration of which method to be used such as calculate average for each group of trials, summarize results in other way or display the data as individual data points. Graphs are often an excellent way to do analysis as display results. There are various purposes of doing charts as data analysis and its relevant chart types as referred to Table 1.3.2.1 shown below.

Purpose of the chart	Appropriate Chart Type
Compare categorical data	Column Chart, Bar Chart, Radar Chart
Compare series of data over time	Area Chart, Line Chart, Column Chart (stacked), High-Low Chart
Percentage of total comparisons	Pie Chart, Donut Chart, Stacked Bar or Column Chart
Relationship between two variables	Scatter Plot
Relationship between three variables	3-Dimension Surface Plot

Table 1.3.2.1: Different Categorization of Charts (Science Buddies 2005-2005)

A good data analysis chart will answer the entire question pertaining to sufficient data to know whether hypothesis is correct, data is accurate, summarizing data with average, specify unit of measurements for all data, and verify that all calculations of the data are correct.

The next thing is to consider the quality of the graph itself. This is where the attribute of the table to generate the graph, such as graph title, independent variables put in appropriate X-axis and dependant variables put in Y-axis, labeling of the axes with measurement units, proper scale and accurate data plotting needs to be considered properly.

System Development

The overall view of the system is depicted in a flowchart diagram (see Appendix Section A for Diagram 1.3.2.2: System Flow)

The system is developed using ASP.NET platform together with IIS. Microsoft first introduced ASP in 1996 and it underwent various versions of updates. It provides a way to create data driven web pages and runs within (IIS). One of the advantages of ASP was that it was relatively easy to be configured on a student's home machine based on Personal Web Server and it would run on Windows from 95 onwards. ASP.NET gives significant advantages over ASP in several ways [Lockyer, Griffiths, Oates, Hebbron, 2003] which are:

- Proper integrated development environment
- Graphical forms based designer
- Integrated debugging
- Use of Microsoft SQL Server

ASP.NET has several features embedded such as WebChart components for developers to render an image in ASP.NET Web applications with user-friendly charting applications.

1.4 Project Timeline

The project timeline consists of two fundamental milestones which involved two semesters in a row for completing this final year project.

The first semester is meant for conducting studies and literature reviews of the project undertaken. A simple non-functioning prototype is being developed in this phase as well. The second semester is the stage for the initial development of the system or product.

The timeline for the first semester and second semester can be referred to Figure 1.4.1 and Figure 1.4.2 in Appendix Section A.

CHAPTER 2

LITERATURE REVIEW AND THEORY

In these literature findings, a survey has been conducted through an interview session with the head of Geo-Chemical Department at PRSS. The responses to open-ended questions can be very useful, often yielding quotable material [Fink, 1995]. In most companies, normal data spreadsheet e.g. Microsoft® Excel or Lotus is widely used as part of data storage and to make data analysis [Muhamad, 2005].

There might be some advantages and disadvantages of using Microsoft® Excel, depends on the situation that might need to use of the spreadsheet program.

The advantages of using Microsoft® Excel lies in fluid power integration [Marsh, 2000] as it considers inherent part of Microsoft® Office suite to many universities, which means many universities, departments and students hold license of this spreadsheet program. The charting and graphing flexibility and also provide for data analysis. Students are more familiar with Microsoft® Excel as a basic for spreadsheets program. Only that they have to configure the formulation need to be input into the specific dialogue box in order to create tables, graphs and charts. Moreover, [Middleton, 1997] indicates that visual display of data in charts and graphs form is very important criteria for data analysis.

A *flat file* [Manheim; Hayes, 2002] is a collection of data organized in columns and rows, pertinent to a given subject. For example, a data table showing various parameters of sample source rock. This is the way the geo-chemical technicians, who often deal with very complex systems, have traditionally collected and analyzed data.

Using spreadsheets like Microsoft® Excel is a popular way to display, manipulate, and create plots and diagrams from flat files. Thus, Microsoft® Excel does provide a chart wizard to create graph from raw data table.

The deficiencies of using Microsoft® Excel lies on the characteristic of the charts or graph that ones need to create. Some of the Excel deficiencies [Cryer, 2001] are pertaining to graphics, help screen, computing algorithms, ands and treatment of missing data. So far, [Cleveland, 1993,1994] and [Tufte, 1983, 1987, 1997] stated that a good presentation of graph or chart considers on clear numerical information, contain no distracting element, appropriate tick marks and label axes, and have descriptive title and/or caption and legend.

For example, the graph types that available as pyramid charts (see Appendix Section A for Figure 2.1: Pyramid Charts). Eventually, the choices of the charts do not represent good graphs as most of the pyramid graph display false third dimensions except the last one. Thus, [Cryer, 2001] stated that this can lead to poor ways to make comparisons because the stacked displays.

Other part that needs to be highlighted as well is the computing algorithms for basic statistics. Excel uses poor algorithms to find the standard deviation, whereby it defines the first quartile to be the ordered observation at position (n+3)/4. Excel does not treat tied observations correctly when it comes to do the ranking and regression computations are prone to error due to poor algorithms.

Currently, there are numerous proprietary software to analyze G & G data. Derived below are a few mentioned.

- Discovery Suite™
    - Discovery Suite™ [Geographix, 2005] is the ultimate Windows® interpretation package gathering all existing data on a prospect or reservoir and translating that information into meaningful, profitable decisions for

the geoscientist. Discovery delivers a whole new level of structural and stratigraphic analysis for geological and geophysical data on the desktop.

- Petris
    - PetrisWINDS Recall PETROS 3 [Petris Technology, Inc., 2005] is the most recent release of Petris' petrophysical analysis package. Major improvements have been made in all areas of data handling and processing, particularly in the data correction, log modeling and interpretation areas. New analysis modules include a comprehensive, yet flexible and easy to use, deterministic log analysis module in which water saturation, net pay and log-derived permeability may be readily set up and run quickly. Full multi-zone, multi-well capabilities, with sophisticated zone and zonal parameter 'spreadsheet-like' editing, have been built in to allow the user to run anything from 'quick-look' single well analyses to large multi-well field studies using the same program.

- ReliaSoft Weibull++7
    - Weibull++7 [ReliaSoft Corporation, 2005] provides a complete array of data analysis, plotting and reporting tools for standard life data analysis (Weibull analysis) with integrated support for a variety of related analyses.

- S-PLUS® 7
    - S-PLUS® 7 Enterprise Developer [Insightful Corporation,2005] was designed to enable statisticians and developers to create targeted statistical applications with large data sets that can be easily deployed to business users, researchers, analysts, and other end users who do not have special expertise in statistical methods.

- Petroleum System
    - The petroleum system is a unifying concept that encompasses all of the disparate elements and processes of petroleum geology. Practical application of petroleum systems can be used in exploration, resource evaluation, and research. A petroleum system encompasses a pod of active source rock and all genetically related oil and gas accumulations.

One of petroleum system presented is the *Mandal-Ekofisk (!)* [Cornford, 1994]. The *Mandal-Ekofisk (!)* petroleum system in the Central Graben of the North Sea, *in* L.B. The petroleum in this system migrated across stratigraphic units (or vertically) into many accumulations. The petroleum system includes the pod of active source rock, the natural distribution network, and the genetically related discovered petroleum occurrences. Presence of petroleum is proof that a system exists. An events chart indicates when the essential elements and processes took place to form a petroleum system, the critical moment, and the preservation time.

However, this system is more towards technical exploration of the well location. It is applicable on the exploration site in which data of the source rock is obtained and being analyzed at the same time during the exploration process.

Although the existing systems can solve problems in most of company pertaining to G&G data management, but from the expert interview referred to [Muhamad, 2005], it is more likely to incurred high cost to obtain commercialize software. The product is already being customized according to general specification of G&G data analysis, thus it does not fit with certain user requirement and provide less flexibility to as tailored to user's need.

CHAPTER 3
METHODOLOGY

3. OVERVIEW

Evolutionary development model has been considered as the methodology for the project. This methodology encompasses of two phases, which are the problem solving identification phase that serve as a basis for software development phase.

3.2 Design Methodology

3.2.1 Evolutionary Development

The evolutionary method is developed to learn about a problem and form a basis for some delivered software [Pfleeger, 2001]. This methodology is idyllic when a user faces uncertainty in what kind of system that he or she requires. Multiple changes maybe requested by the user, which could lead to delay of the whole project. However, this methodology has the ability to cater such issue by performing concurrent phases and allow rhythmic tasks to happen.

One of the fundamental kinds of evolutionary development [Sommerville, 2004] is the exploratory development where the objective of the process is to work with the customer to explore their requirements and deliver a final system. The development starts with the parts of the system that are understood. The system evolves by adding new features proposed by the customer.

Evolutionary method has been used by Hewlett Packard [May; Zimmer, 1996] and the project postmortem listed a number of benefits from using evolutionary development:

- Long-term vision broken into short-term steps
- Prioritized implementation within component teams
- External customer feedback
- Early realism about how much can be done.

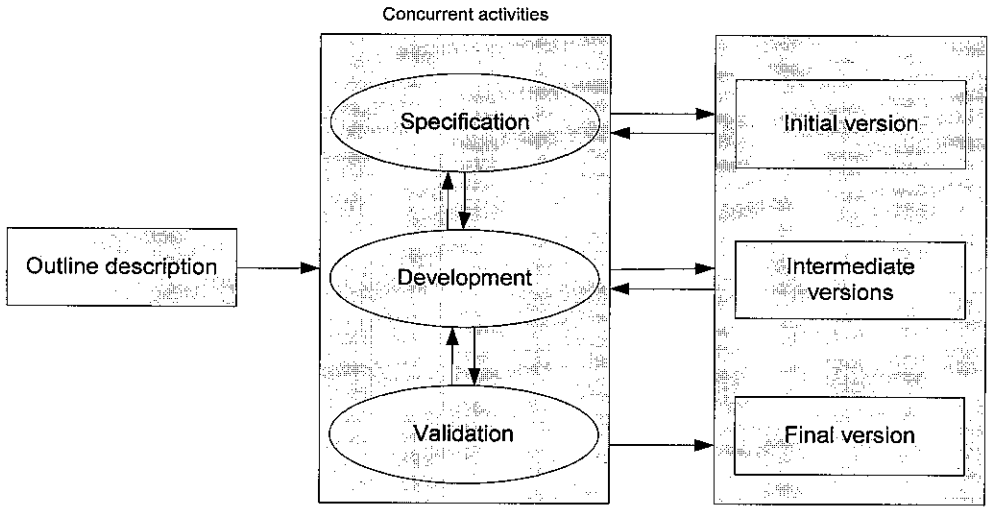


Figure 3.2.1.1: Evolutionary Development

The evolutionary development is based on idea of developing initial implementation, revealing this to user comment and refining it through many versions until an adequate system has been developed (Figure 3.2.1.1). Specification, development and validation activities are interleaved rather than separate, with rapid feedback across activities.

- Specification
    - The concept is to have an online system that provides data analysis in chart form of geo-chemical data. The requirement for the system is to assist geo-chemist and higher management of Geo-chemical Department in dealing with massive geo-chemical data and to generate quick and near to accurate data analysis charts for them to locate oil and gas well.

- Development
    - The system will be using ASP.NET as the platform, Microsoft SQL server as its database and IIS to enable an online system. The essential part of the system is to have chart components, which are available in ASP.NET platform. The system will go through a complete design phase as to verify on the essential control structure, interface design and modules of the system.

- Validation
    - The system will integrate a properly function modules, which composed of, chart data analysis that can be generated automatically and online system with geo-chemical manipulation. This is also to check whether the working functions meet with user requirements.

3.3 Tools

The required tools to develop this online automated system are:

- Microsoft Visual Studio.Net framework.
- ASP.NET as the platform with VB language.
- Microsoft SQL server as the database platform.
- WebChart Control.

For the charting part, the ASP.NET provides a component which is called as WebChart Control. WebChart Control is relatively easy-to-use and it's free component to create a variety of common charts. For this purpose, the component can be installed within ASP.NET framework. To be compared to other commercialize chart component, WebChart Control provides simple chart for small web application that incur in low cost.

CHAPTER 4

RESULTS AND DISCUSSIONS

4. OVERVIEW

This section will discuss and summaries the literature findings and the system process flow. The fundamental parts of the online automated data analysis system are to give near to accurate results and generate a quick charting process. Based on the literature review, the spreadsheet program had been the effective tool for generating charts and graphs. The existing and available data analysis system is found to be costly and complex to be handled.

The system developed would allow flexibility in terms of user requirements and ease of use. One of the advantages offered by the system is that it does not require any facet knowledge on the subject matter while handling the system. The system is equipped with several levels of authentications depending on user's privilege. For example, the geo-chemist is allowed to have full access to the data while clients will be allowed to access a portion of data depending on their privileges. Hence, guarantee the overall integrity of the system and data.

4.1 Results

The first objective of the project, which is to study on how data analysis is perform when dealing with massive geo-chemical data have been met. All the fact findings regarding G&G data analysis have been carried out through interview session and studies.

The second objective has been met. The system is fully functional and has been made online which available at *http://163.0.2.62/OnlineGnG/Login.aspx* via UTP local area network. The system can be run online by changing the IP number in the URL link according to one computers IP address. The discussion part will be touching over the system's overview, system's flow.

4.2 Discussion

4.2.1 Interview Session

An interview session has been conducted with personnel from Geoscience group at PRSS. The questions are related to how the department deals with massive geo-chemical data, as well as to investigate the current system in use.

From the interview results, it shows that the geo-chemical technicians have to take lots of time or up to several days to manually plot on single graph. This can contribute to the delay of the department reporting progress.

4.2.2 Research

The study is done based on the information available on online journals. Fact-findings consist of the related G&G data analysis function, current system available and types of charts. Further studies revolve around the platform that will be used for development, such as ASP.NET and MSQL Server components.

Performing data analysis is very essential in giving interpreting oil and gas well location. Generating charts and plotting graphs that is capable in resulting near to accurate interpretation is essential for this purpose. Thus, types of charts such as line

chart, histogram and other related graph is necessary for data presentation in visualize form and to determine which is appropriate to be used.

The use of spreadsheet has its own drawbacks. The normal spreadsheet has the function to generate charts, but still have limitation in some area for the geo-chemical technician to create the charts in terms of integration with massive data and less accurate of plotting chart.

4.2.3 System Overview

From evolutionary method, basic functionality of the system has been identified. Thorough description is being outlined in Table 4.2.3.1 as shown below.

No	Module	Description
1	**Data Entry and Manipulation**	• *Manage the input for G&G data into the system database* • *Query and Retrieve* o To query G&G data pertaining to its information such as date of sample data taken and well location.
2	**Data Processing**	• *Generate Graphs/Charts* o To generate graph/chart function driven by the G&G data stored in database. o The generated graph/chart will give the results of interpretation of potential oil and gas well location.
3	**Reporting**	• *User Report Viewing* o Report generated will be in tabular region that will hold all the generated graphs/charts related to G&G data analysis.

Table 4.2.3.1: List of module available in the system

4.2.3.1 The System's Use Case Diagram

The system use case diagram can be referred to Figure 4.2.3.1.1 in Appendix Section A. The use case diagram described what kind of users involved directly with the system. The users of the system are listed below:

- Administrator
  - The Administrator is the one who had developed the system. Thus, the administrator is also the one who will have the highest privileges to manage the whole system such as to add and delete user directly from the system database.

- G & G Technician
  - The personnel or G & G technician from the Geo-Chemical Department will be given the privileges of managing the data such as to do the oil well data entry, view well's data table and generating charts. Of course, this authentication level will not allow adding or deleting user from a database.

- Normal User
  - The normal users are range from personnel from various locations such as from different department to view the charting reports. To ensure integrity of the system, the new user should register first to enroll to the online system.

4.2.3.2 The System's Process Flow

The process flow is essential in describing how the system navigation will look like. The system flowchart (see Figure 4.2.3.2.1 in Appendix Section A) is depicted to show the overall process of the online system that has been developed.

The first module will include user access through the login page (see Figure 4.2.3.2.2 in Appendix Section B) and well information data entry. The login access will cater for administrator and normal user. Once the user logged in, the page will be directed to the second page. The navigation link menus are resided in the i-frame at the left panel for user to select (see Figure 4.2.3.2.3 in Appendix Section B).

For the second and third modules, it combines in the navigation links which consist of Import Old Data, Manage New Data, View Well's Tables, and Generate Charts.

- Import Old Data:
    - The old existing data will not be used anymore but this module is mainly for transferring data from Excel file into a new database system. The old data will be preserved for viewing purpose only. (See Figure 4.2.3.2.4 until Figure 4.2.3.2.7 in Appendix Section B)

- Manage New Data:
    - The user will need to enter new well information and well details information. The data entry is useful for the purpose of generating chart later on as the process is continuous. The general well information is the first data entry that contains the well's name, operator, block and depth Once completed, the user can continue to view the data table and then from the table itself, there is a button indicating for 'add detail' to proceed with the oil well's detail data entry. After successfully added, the user can repeat the process of this data entry according to the data that need to be entered at one particular time. Note that this oil well

detail data entry is meant for the one particular well that have been entered during the first data entry. For example, if the user had entered a new oil well named 'Anding Barat' then the user can proceed to second data entry to add more details about this 'Anding Barat' oil well. (Refer to Figure 4.2.3.2.8 - Figure 4.2.3.2.15 in Appendix Section B)

- View Well's Tables:
    - This module is to cater the purpose of user who might want to view only general well's information such as the name of the well whereby the data had been entered before. Thus, the viewed data table consists of button indicating the same direction for the second data entry, which is to add oil well's detail. (See Figure 4.2.3.2.16 and Figure 4.2.3.2.17 Appendix Section B)

- Generate Charts:
    - Subsequently the main function of the system is to generate chart from the data that have been entered before. The data table containing the well's name will be the key of which oil well to be selected. Upon selecting particular well, the oil well detail's data table can be viewed and have the button indicating 'Select as Start Row'. This is to make sure where the chart should set its first value before plotting. The same goes to the next procedure whereby the user will have to select for end row to indicate the last value in chart plotting. When user is done, the button generate chart can be clicked and the system will automatically generate the chart. The chart will be generated according to the data that have been selected from the dropdown lists for the x and y axis value.
    - The interpretation from data analysis to locate potential oil well location will be done during this process. The data interpretation is categorized according to the value ranges at X-axis and divided into minimum, low, median and high value. From geochemical parameters, the plotting is useful for determining the good quality of source rocks. The

interpretation part will accord to the range of value; starting from minimum value from the sample G & G data taken. The lowest range indicates as immature oil, medium range indicates oil window and highest range indicates gas window. The possible oil or gas window interpretation is helpful in concluding oil or gas reservoir later on by mapping on the targeted oil and gas reservoir areas. (See Figure 4.2.3.2.18 until Figure 4.2.3.2.24 in Appendix Section B).

- For saving the current generated chart, the button 'Submit' will direct user to the new page where they have to enter chart's title. The 'Save' button is click to save the entire chart's information as new record in database. The chart is saved by sending parameters according to the chart's title and user login into the database. The same way is done to display back the chart image as the database does not support for saving image. (See Figure 4.2.3.2.25 – Figure 4.2.3.2.29 in Appendix Section B).

The system coding is also being attached in Appendix Section C.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5. Overview

Current massive sedimentary data obtained by PRSS from exploration site is somehow less organized. Confusion usually happens when all the data is being bulked together, be it analyzed or not. Due to this lack of data organization, the manually charting process is affected. Therefore, the analysis part is less accurate, thus results in imprecise prediction of the well location. Hence, it is vital to have an online automated system that is capable of handling immense data of various form, which allows a near to accurate data analysis based on automated charting and graphing provided with itinerant features.

5.1 Conclusion

A thorough study on how data analysis is done when dealing with massive geo-chemical data had been conducted. Interview session with experts yield valuable information revealing the problems they faced in this area. Literature review, journals and articles searched reveals previous work on crucial elements of the subject matter. These fact-findings methods contribute to better understanding by providing references to conduct the research.

The online automated system is a system that can generate charts as well as giving interpretation for oil well location. The system integrates data management in order to

maintain the oil well's information to make it easy for the user to keep track on particular oil well information from time to time and for future references. This is referring to the user who logged in o the system and the date of particular data had been entered.

The process of generating chart is quite flexible in a way that the system provides user selection for the start row and end row from the oil well data table. This is much easier to do the charting process whereby the data can be very massive and not all can be selected for each X-axis and Y-axis values.

The online system will make it easy for the other user from other department can also viewed the charting report as part of the elements in decision-making of oil well exploration. Thus, it takes less time for the higher management to search for the report in terms of paper environment as the system will be just one click away from their fingertips.

5.2 Recommendation

Throughout this whole project, there are several recommendations for the system that is potentially to be developed as the system still has some limitation such as plotting in 2-D chart and data interpretation.

- Integration with GIS
    - The online automated system is recommended to be integrating directly with GIS system, which resides in centralized database. Thus, any information pertaining to well location from geo-chemical data can be retrieved in visualize manner. For example, the location can be viewed through the map when user select the specific well name desired from the system that have stored all the necessary well's information.

- Security
    - The level of authentication for the system should be enhanced. The authentication is according to user's privileges to have access to the online system. Further security measure is through encryption. The privacy of communications between the client's browsers and the system's server is ensured via encryption. When an external client gets access to the system, the browser should establish a secure session with the system's server. Users will be unable to access the online system functions at lesser encryption levels, depending on the server encryption requirement. This may require some end-users to upgrade their browser to a stronger encryption level.

- Conduct Test
    - Several tests are to be conducted to gain feedback from many respondents who will use the system in future. This is to ensure the system integrity and its heftiness to give better results in data analysis. The test is also to keep track of any error that might occur while using the system. The type of test conducted will be of white box and black box testing. The white box testing is meant for expert or advance user to test upon the system, whilst the black box is for novice user who doesn't have a depth knowledge regarding how to handle the system.

- Accuracy
    - Improve accurate data analysis leads to better prediction of potential petroleum well location. This is due to the massive geo-chemical data parameters, which have to be plotted into chart or graph to give analyzed results. In accordance to oil and gas well exploration, it is optimism for the system to be enhanced with seismic data interpretation.

5.1.1 Recommendation from External Examiner

As part of the external examiner's recommendation for the system, it is possibly essential to develop such system into a mobile application. For instance, the mobility of the system can contribute to better ubiquitous access by higher management personnel. The mobile devices such as cell phone or Personal Digital Assistant (PDA) can be used by the personnel involved in decision making during a meeting despite of where they are located.

The mobile application also can integrate with GIS as the mobile also support for GIS application. Thus, the targeted oil well location can be viewed in visualized manner.

REFERENCES

Bhatia, M. R. (1983), *Plate tectonics and geo-chemical composition of sandstones,* Journal of Geology, vol. 91, 611-649.

Card, D.N. (2004), *Statistical Techniques for Software Engineering Practice,* Software Productivity Consortium. Retrieved October 12, 2005, from http://nr.stic.gov.tw/ejournal/ProceedingD/v8n3/108-120.pdf

Cleveland, W. S. (1993), *Visualizing Data*, Hobart Press, Summit, NJ

Cleveland, W. S. (1994), *The Elements of Graphing Data*, Revised Edition, Hobart Press, Summit, NJ

CMMI Development Team (2001), *Capability Maturity Model – Integrated, (Version 1.1)*, Software Engineering Institute.

Cornford, C. (1994), The Mandal-Ekofisk(!) petroleum system in the Central Graben of the North Sea, *in* L.B. Magoon and W.G. Dow, eds., The Petroleum System--From Source to Trap: AAPG Memoir 60, p. 537-571.

Cryer, J.D. (2001), *Problems with Using Microsoft Excel for Statistics*, Joint Statistical Meetings, Department of Statistics and Actuarial Science, University of Iowa. Retrieved October 12, 2005, from http://www.stat.uiowa.edu/~jcryer/JSMTalk2001.pdf

Fedo, C. M., Nesbitt, H. W. and Young, G. M. (1995), *Unraveling the effect of potassium metasomatism in sedimentary rocks and paleosols, with implications for paleoweathering conditions and province*, Geology, vol. 23, no. 10, p. 921-924.

Fink, A. (1995), *How to ask survey questions*. Thousand Oaks, CA: SAGE Publication

Garrett, R.G., Reimann, C and Filzmoser, P. (2004), *Multivariate outlier detection in exploration geochemistry*, Computers & Geosciences 31 (2005), 579–587.

Geographix (2005), Geographix Company's Site. Retrieved October 12, 2005, from http://www.geographix.com

Insightful Corporation (2005), Insightful Compay's Site. Retrieved October 12, 2005, from http://www.insightful.com

Kiminami, K., Kumon.. F., Miyamoto, T., Suzuki, S., Takeuchi, M. and Yoshida, T., (1998), *Application of B. I. diagram to Paleozoic and Mesozoic sandstones of the Japanese Islands: Sandstone Composition and Tectonics of East Asia Tectonic Zone*, Ed. Hoyanagi, K., no. 4, p. 1-7

Lacey, A. and Luff, D. (2001), *An Introduction to Qualitative Analysis, Trent Focus for Research and Development in Primary Health Care*, Trent Focus.
Retrieved from October 12, 2005 from
http://www.trentfocus.org.uk/Resources/Qualitative%20Data%20Analysis.pdf

Lockyer, M., Griffiths, G., Oates, B., Hebbron, B. (2003), *.NET & Web Services, School of Computing,* University of Teesside. Retrieved October 12, 2005, from http://www.xtras.net/products/componentonewebchartforaspdotnet

Manheim, F.T., and Hayes, L. (2002), *Sediment database and geochemical assessment of Lake Pontchartrain Basin,* Chapter J of USGS Professional Paper 1634. Retrieved October 26, 2005, from http://pubs.usgs.gov/prof/p1634j/idex.htm#contents

Marsh, B. (2000), *Integrating Spreadsheet Templates and Data Analysis into Fluid Power Instruction,* Journal of Industrial Technology. Retrieved October 12, 2005, from http://www.nait.org/jit/Articles/marsh071900.pdf

May, E.L. and Zimmer, B.A. (1996), *The Evolutionary Development Model for Software*, Hewlett-Packard Journal. Retrieved October 26, 2005, from http://www.hpl.hp.com/hpjournal/96aug/aug96a4.pdf

Middleton, M. (1997), *Data analysis using Microsoft Excel.* Belmont, California: Wadsworth Publishing Co.

Muhamad, A.J. (2005, October 4), Geosciences Group, *Personal Interview*. Petronas Research Scientific & Services, Bangi.

Petris Technology, Inc. (2005), Petris Company's Site. Retrieved from October 12, 2005, from http://www.petris.com

Pfleeger, S.L. (2001), *Software Engineering Theory and Practice*. 2nd. ed. Prentice Hall International, Inc, p. 169

Reimann, C., Filzmoser, P., Garrett, R.G., (2005), *Background and threshold: critical comparison of methods of determination*, Science of the Total Environment, in press.

Reliasoft Corporation (2005), Reliasoft Company's Site. Retrieved October 8, 2005, from http://www.reliasoft.com

Sanchez, P. (2005), *The Benefits of using Stored Procedures*. BlueOak Database Engineering White Paper.

ScienceBuddyies. (2002-2005), *Data Analysis & Graphs*. Retrieved October 12, 2005, from http://www.sciencebuddies.org/mentoring/project_data_analysis.shtml

Sommerville, I. (2004), *Software Engineering*. 7th. Ed. Addison Wesley, p. 68 - 71

Technical Committee (2000), *ISO Standard 9001, Quality Management Systems*, International Organization for Standardization.

Tufte, E. R. (1983), The *Visual Display of Quantitative Information*, Graphics Press, Cheshire, Conn.

Tufte, E. R. (1990), *Envisioning Information*, Graphics Press, Cheshire, Conn.

Tufte, E. R. (1997), *Visual Explanations*, Graphics Press, Cheshire, Conn.

APPENDIX SECTION A

Petroleum Potential	Organic matter			Bitumen		Hydrocarbon
	TOC (wt.%)	S_1[a]	S_2[b]	(wt.%)	(ppm)	(ppm)
Poor	0-0.5	0-0.5	0-2.5	0-0.05	0-500	0-300
Fair	0.5 -1	0.5-1	2.5-1	0.05-0.10	500-1000	300-600
Good	1-2	1-2	5-10	0.10-0.20	1000-2000	600-1200
Very good	2-4	2-4	10-20	0.20-0.40	2000-4000	1200-2400
Excellent	>4	>4	>20	>0.40	>4000	>2400

[a]mg HC/g dry rock distilled by pyrolysis
[b]mg HC/g dry rock cracked from kerogen by pyrolysis

(from L.B. Magoon , 1994)

Figure 1.3.2.1: Parameters of Immature Source Rock

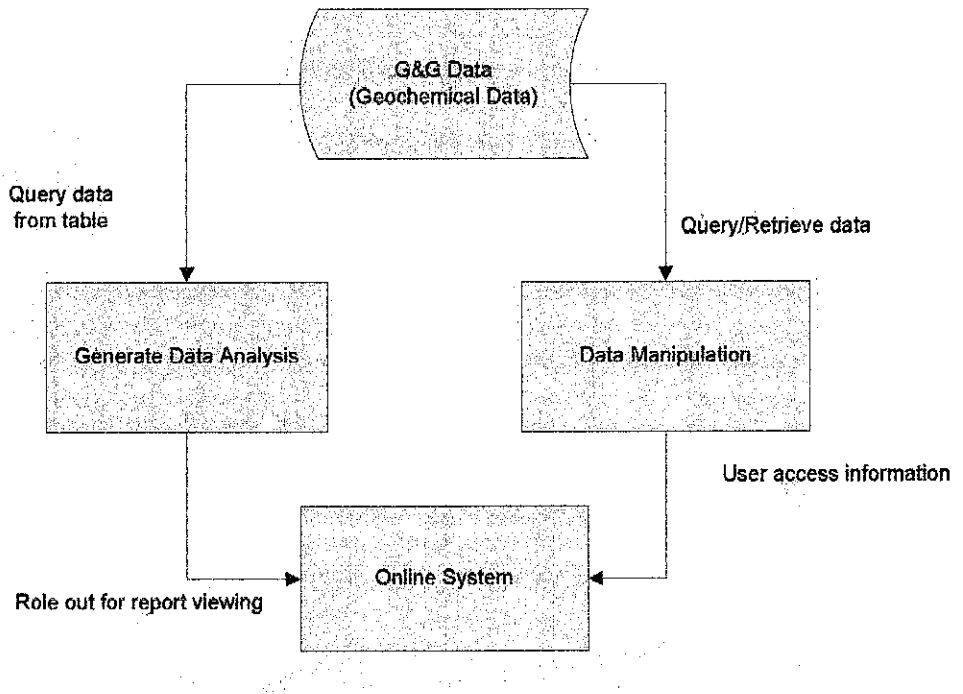


Figure 1.3.2.2: General System Flow

Project Schedule for FYP Part I

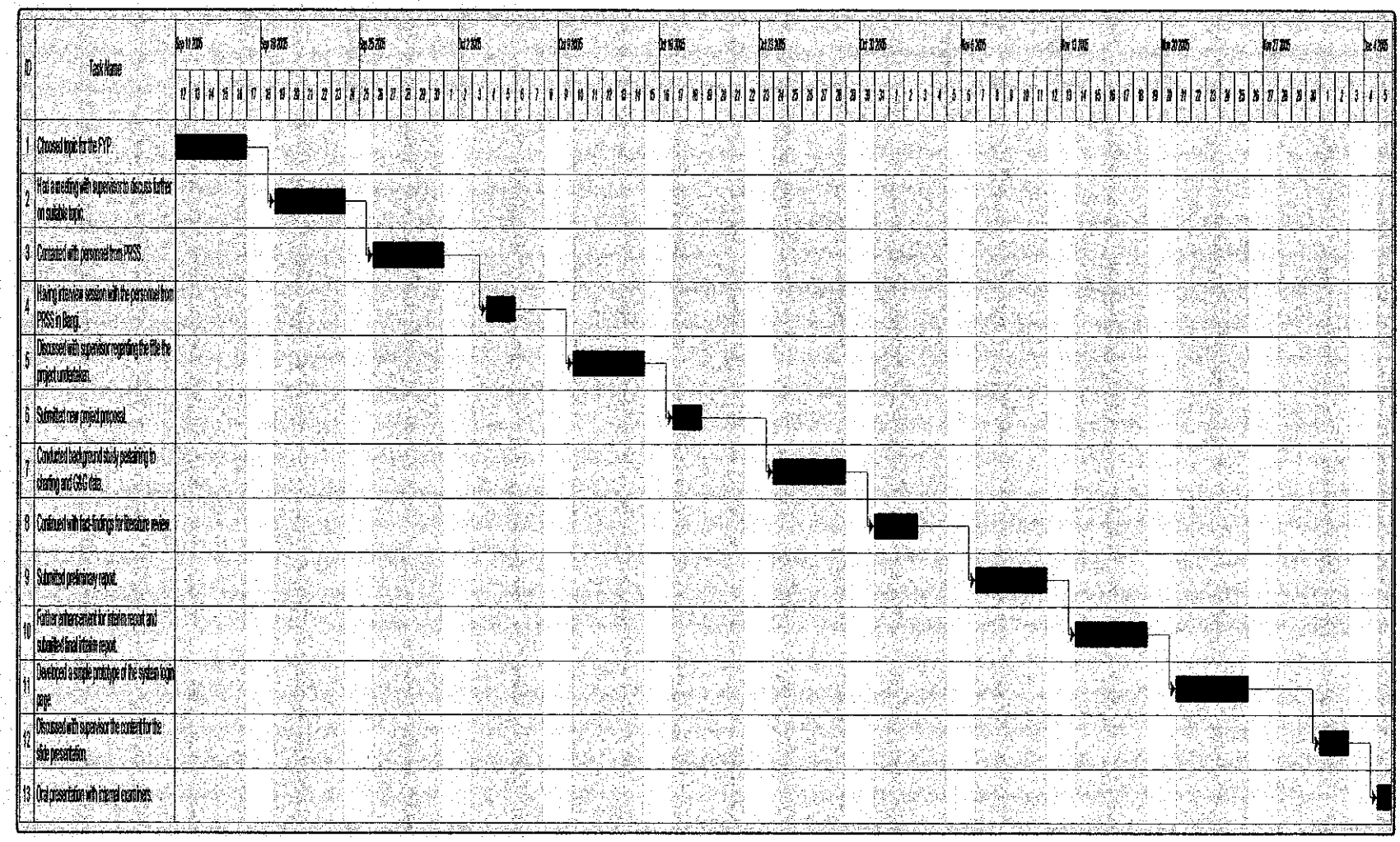


Figure 1.4.1: Gantt Chart for FYP Part I

Project Schedule for FYP Part II

ID	Task Name	Start	Finish	Duration
1	Reviewed and enhance the system's design.	1/23/2006	1/27/2006	1w
2	Browsed through the internet to download webchart component.	1/30/2006	2/3/2006	1w
3	Meeting with supervisor for the content of progress report.	2/6/2006	2/10/2006	1w
4	Continued with designing user interface.	2/13/2006	2/17/2006	1w
5	Added i-frame for navigation links of the online system and create database.	2/20/2006	2/24/2006	1w
6	Design on data management module.	2/27/2006	3/3/2006	1w
7	Develop a module for importing excel data into the system's database.	3/13/2006	3/17/2006	1w
8	Continued with data entry module foe oil well information.	3/20/2006	3/24/2006	1w
9	Meeting with supervisor and altered some of the system's interface.	3/27/2006	3/31/2006	1w
10	Done for generating chart module and prepared for Pre-EDX presentation.	4/3/2006	4/7/2006	1w
11	Studied on statistical formula through the internet and try to revise other calculation on how to interpret data.	4/10/2006	4/14/2006	1w
12	Proceed with the variation of webchart control component	4/17/2006	4/21/2006	1w
13	Submitted final report to the external examinier.	5/8/2006	5/12/2006	1w
14	Oral presentation with internal and external examiners.	6/19/2006	6/23/2006	1w

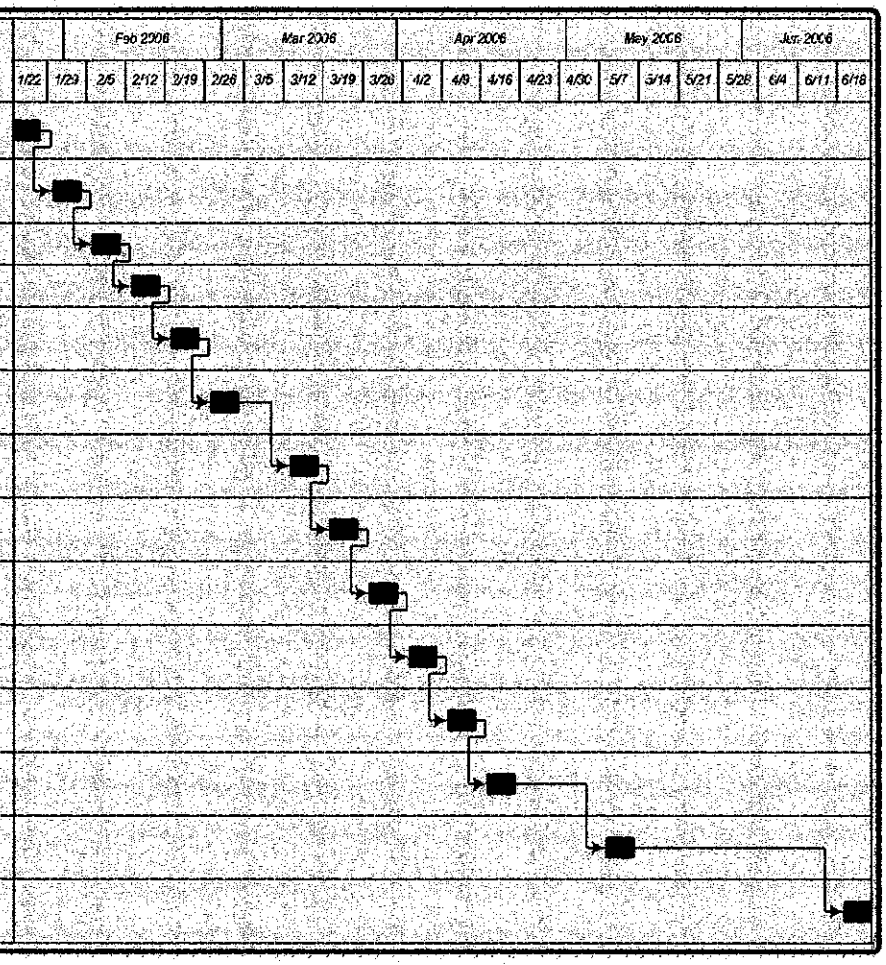


Figure 1.4.2: Gantt Chart for FYP Part II

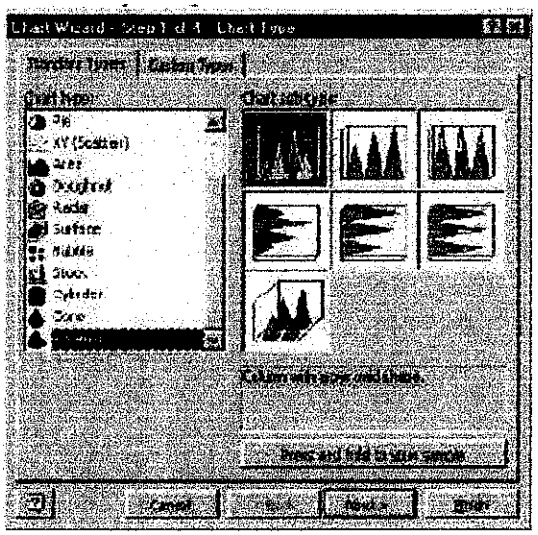


Figure 2.1: Pyramid Charts

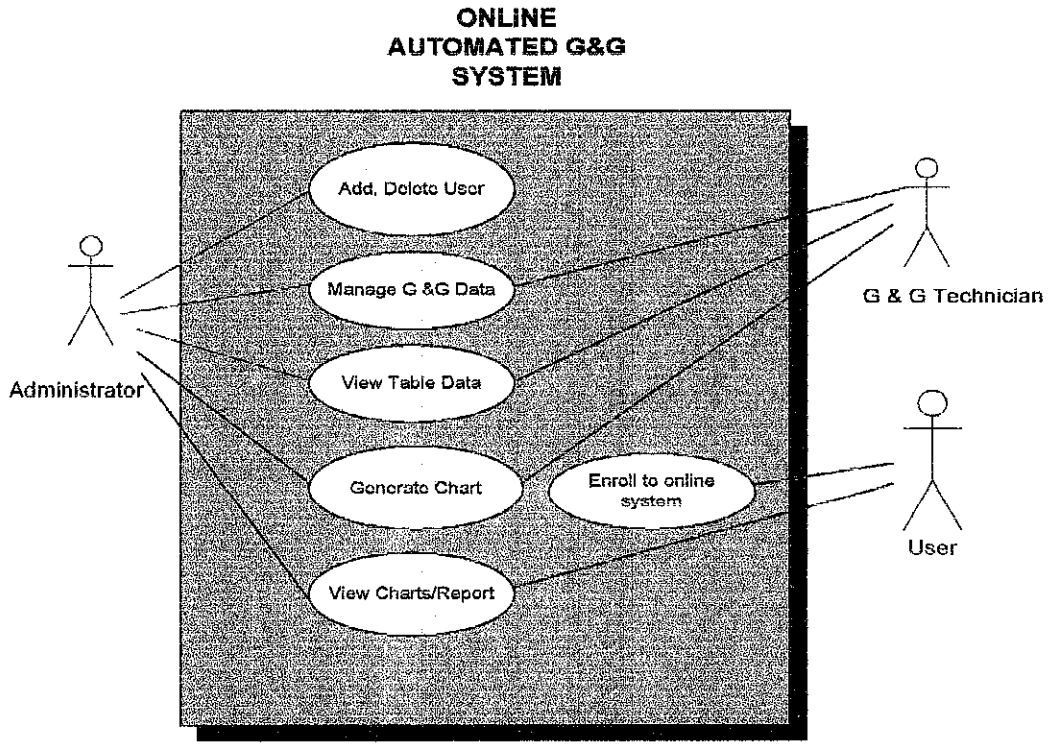


Figure 4.2.3.1 Use Case Diagram of Overall System Process Flow

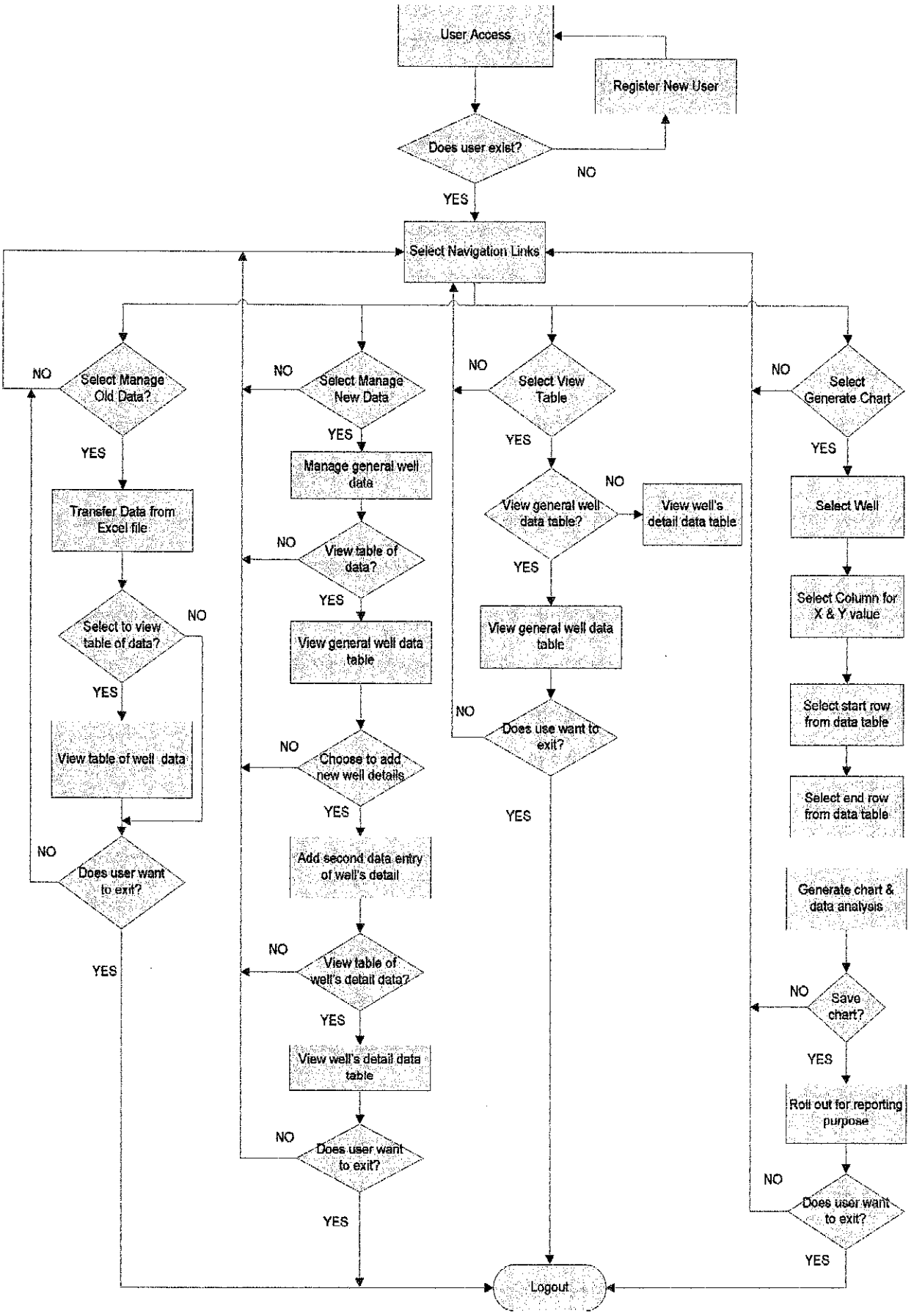


Figure 4.2.3.2.1: Process Flow

APPENDIX SECTION B:
SYSTEM'S SCREEN SHOTS

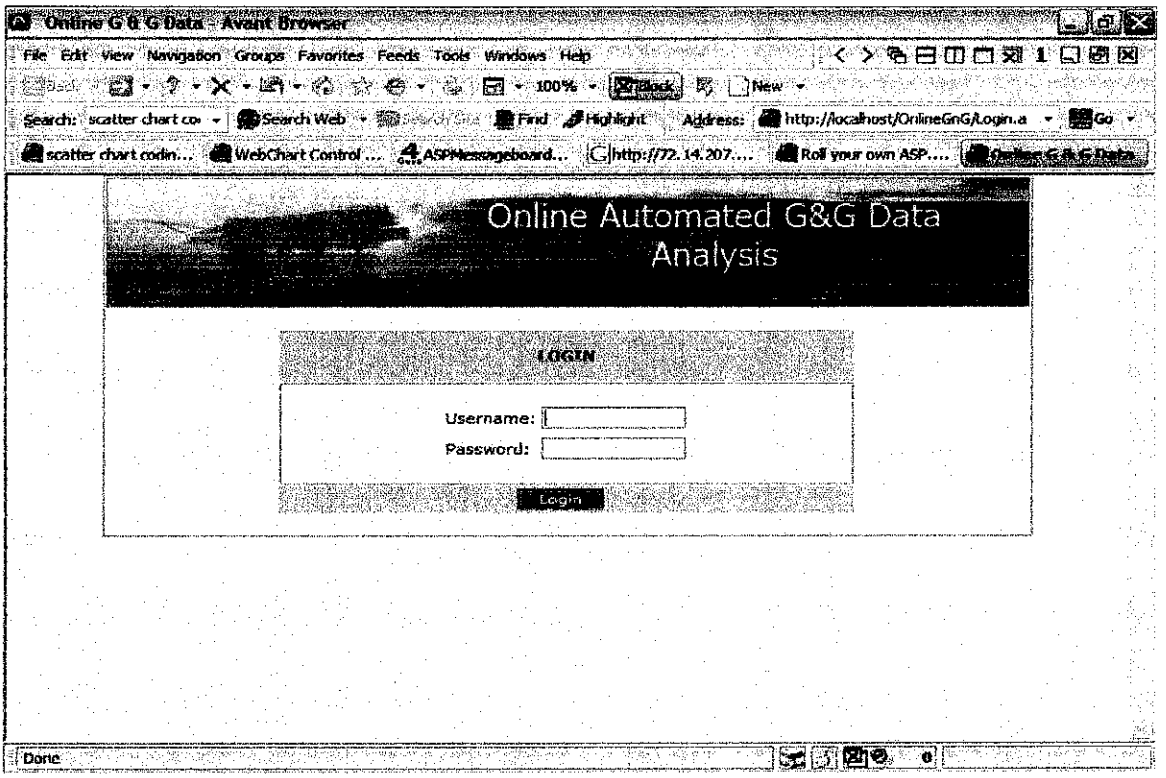


Figure 4.2.3.2.2: Login page

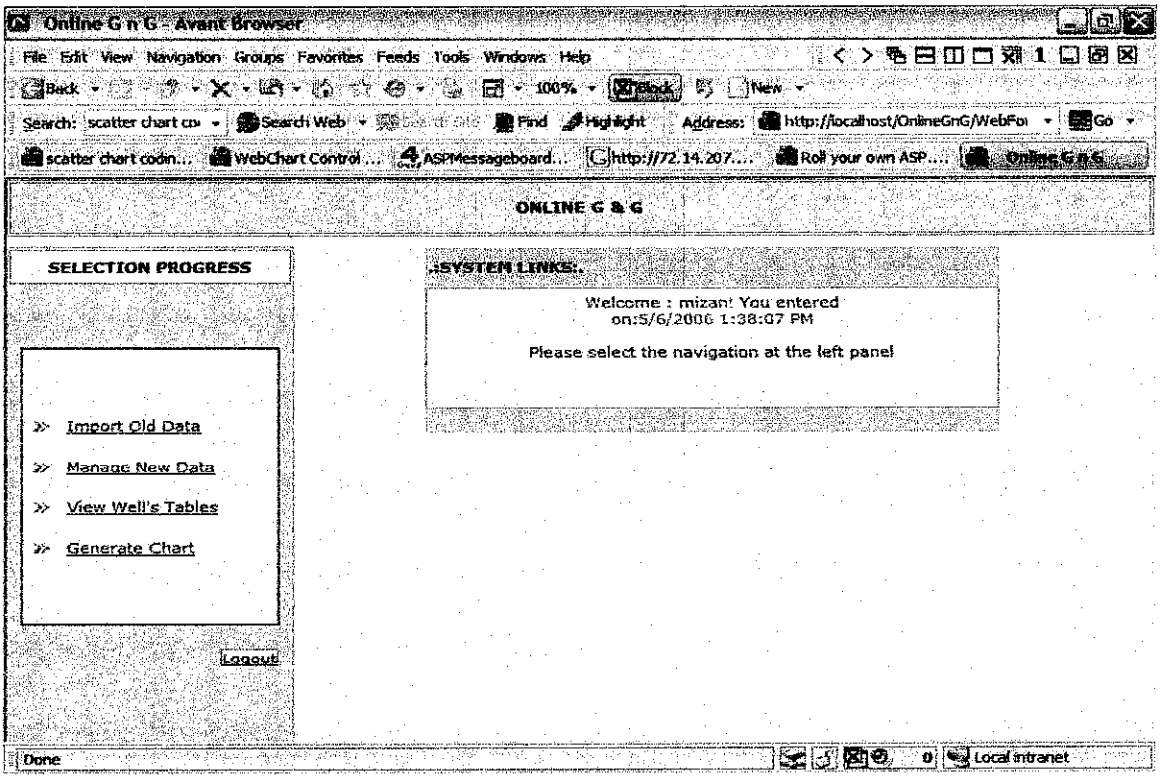


Figure 4.2.3.2.3: Welcome page

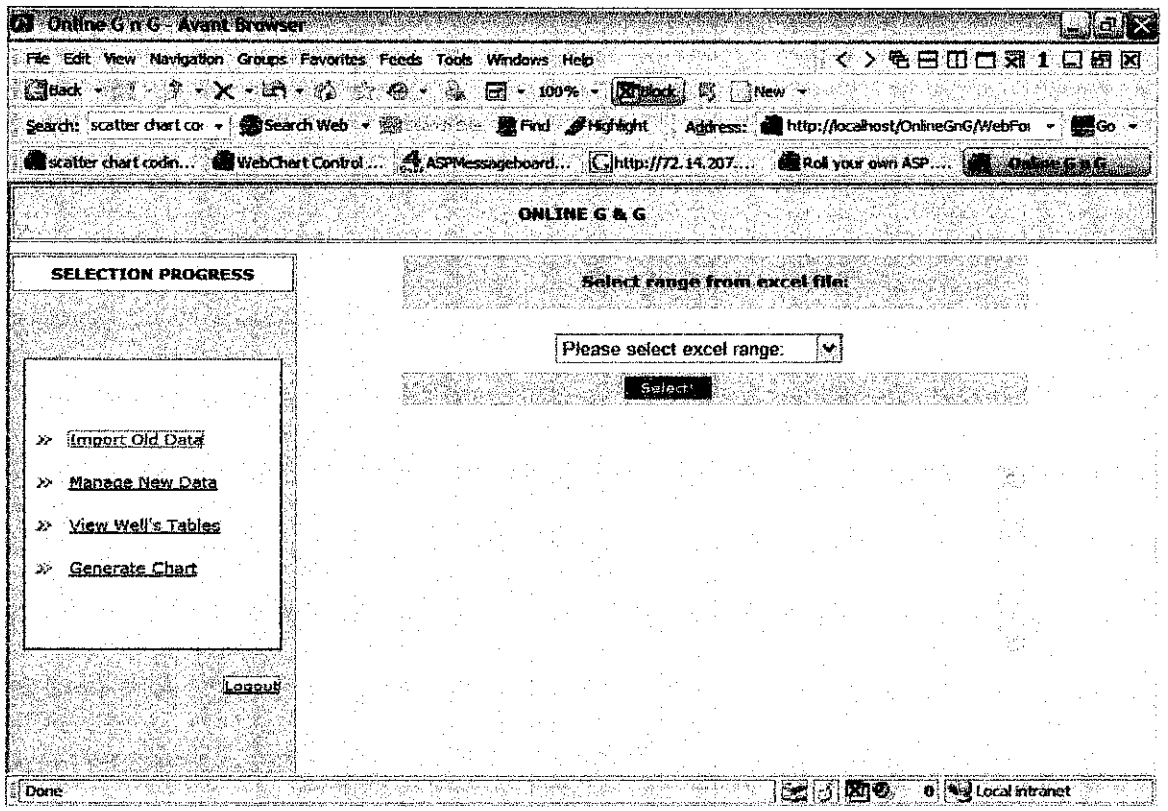


Figure 4.2.3.2.4: Import Excel data page

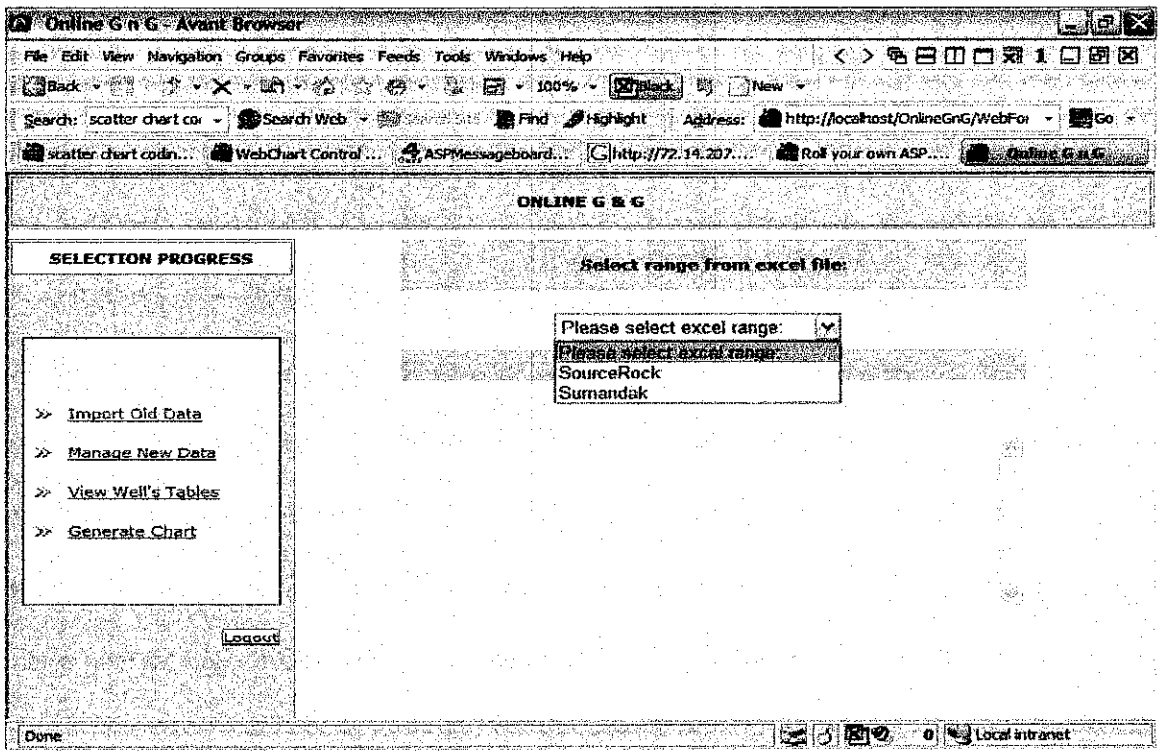


Figure 4.2.3.2.5: Select Excel file

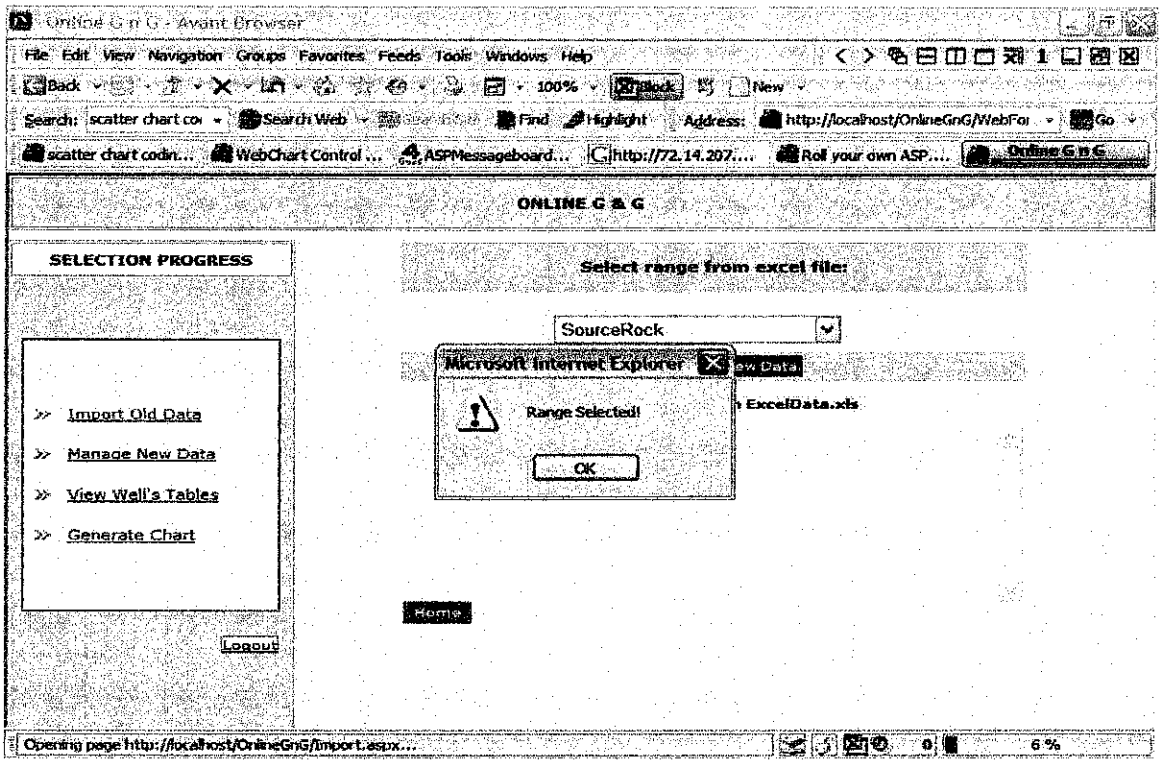


Figure 4.2.3.2.6: Pop-up message indicating selected range of Excel data

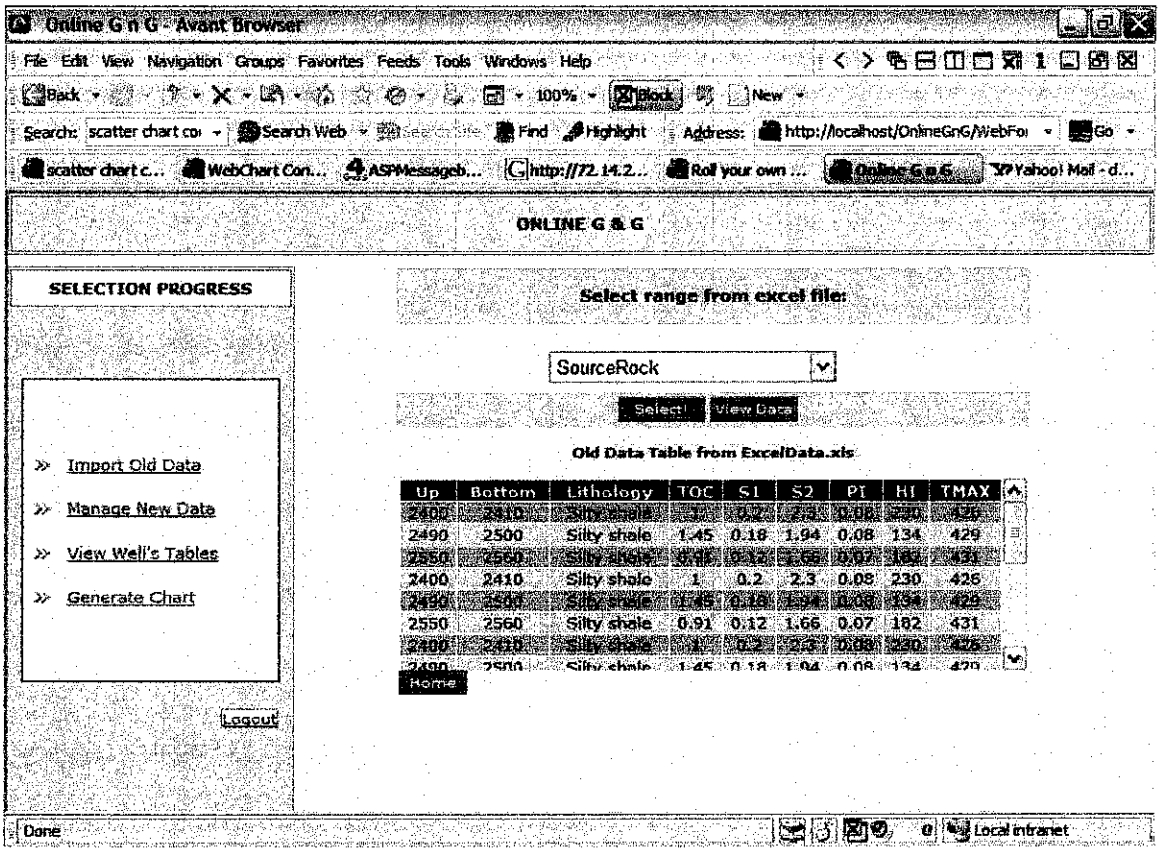


Figure 4.2.3.2.7: View Excel data in table Form

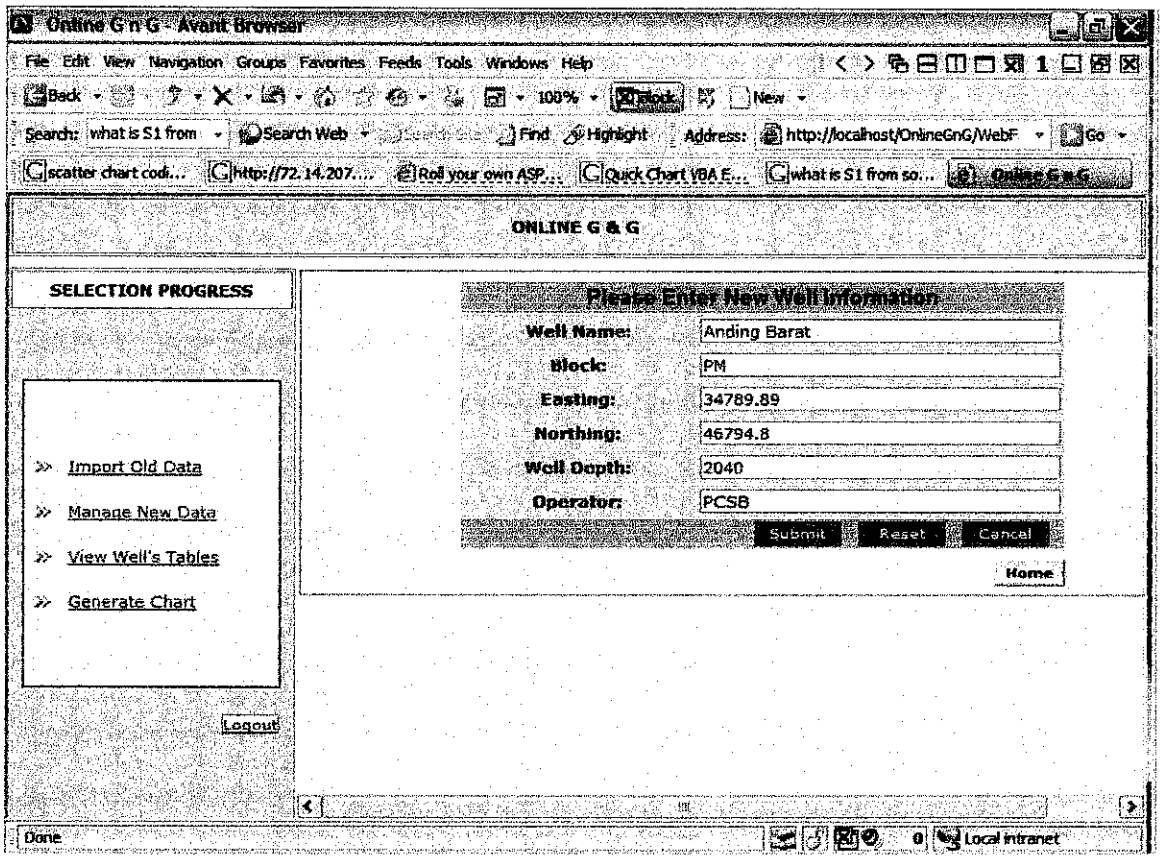


Figure 4.2.3.2.8: Data entry for general well information

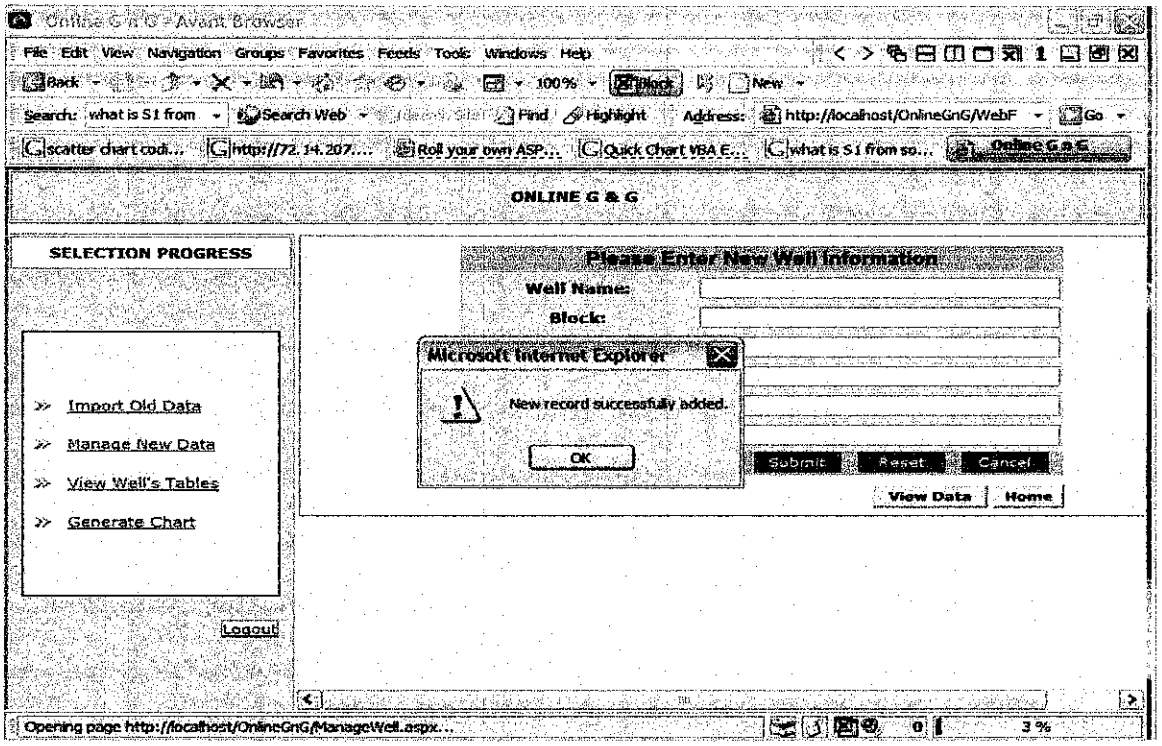


Figure 4.2.3.2.9: Pop-up message indicating successful record entry

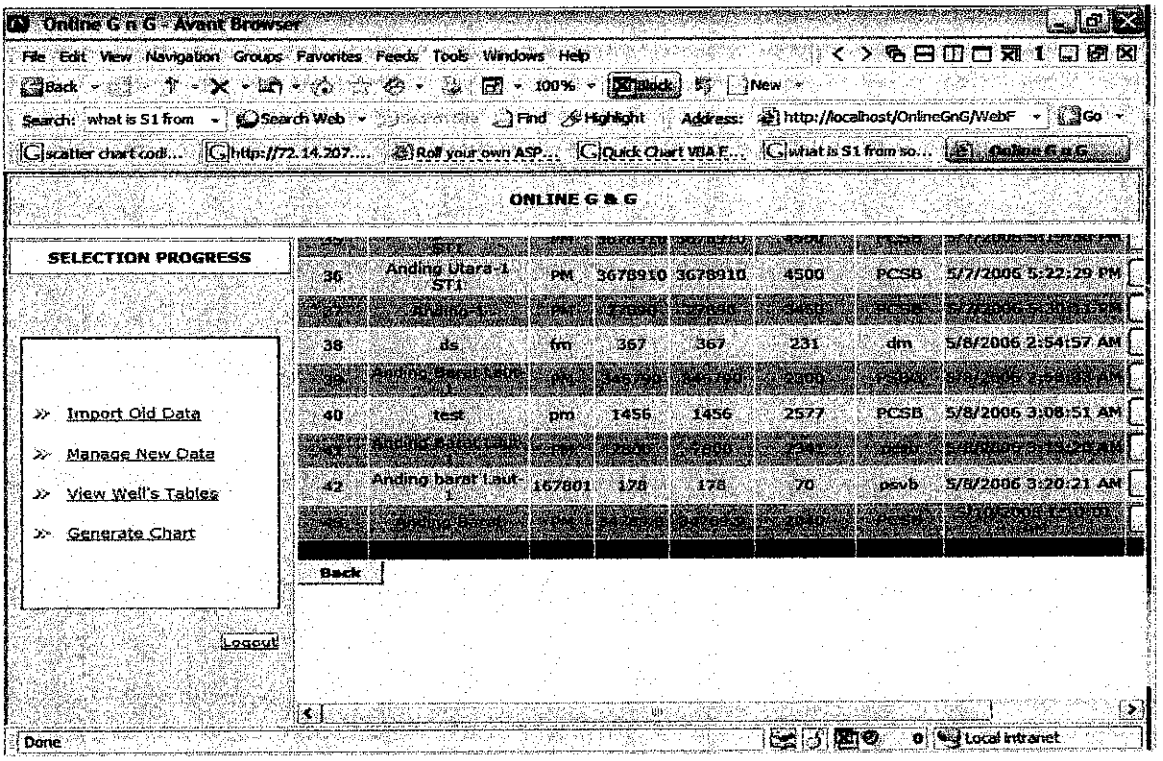


Figure 4.2.3.2.10: View general oil well data table

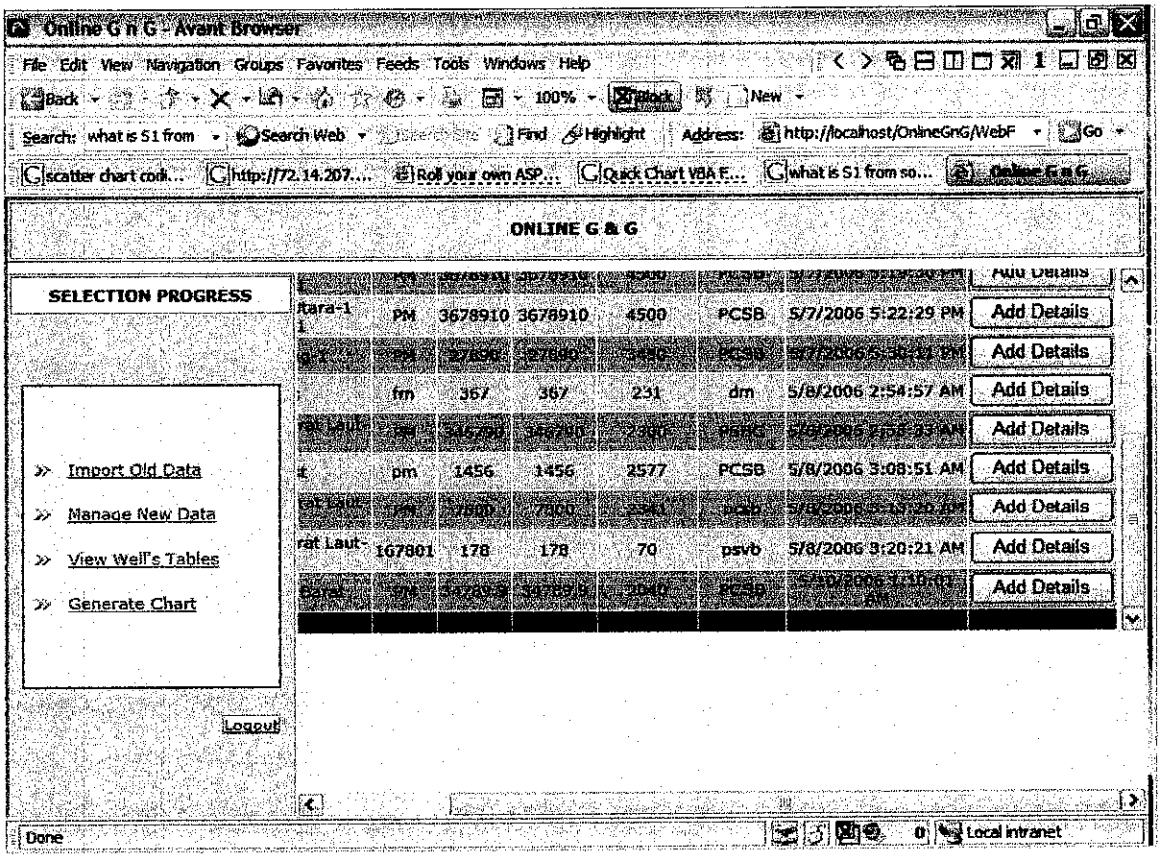


Figure 4.2.3.2.11: Button option for adding oil well's detail

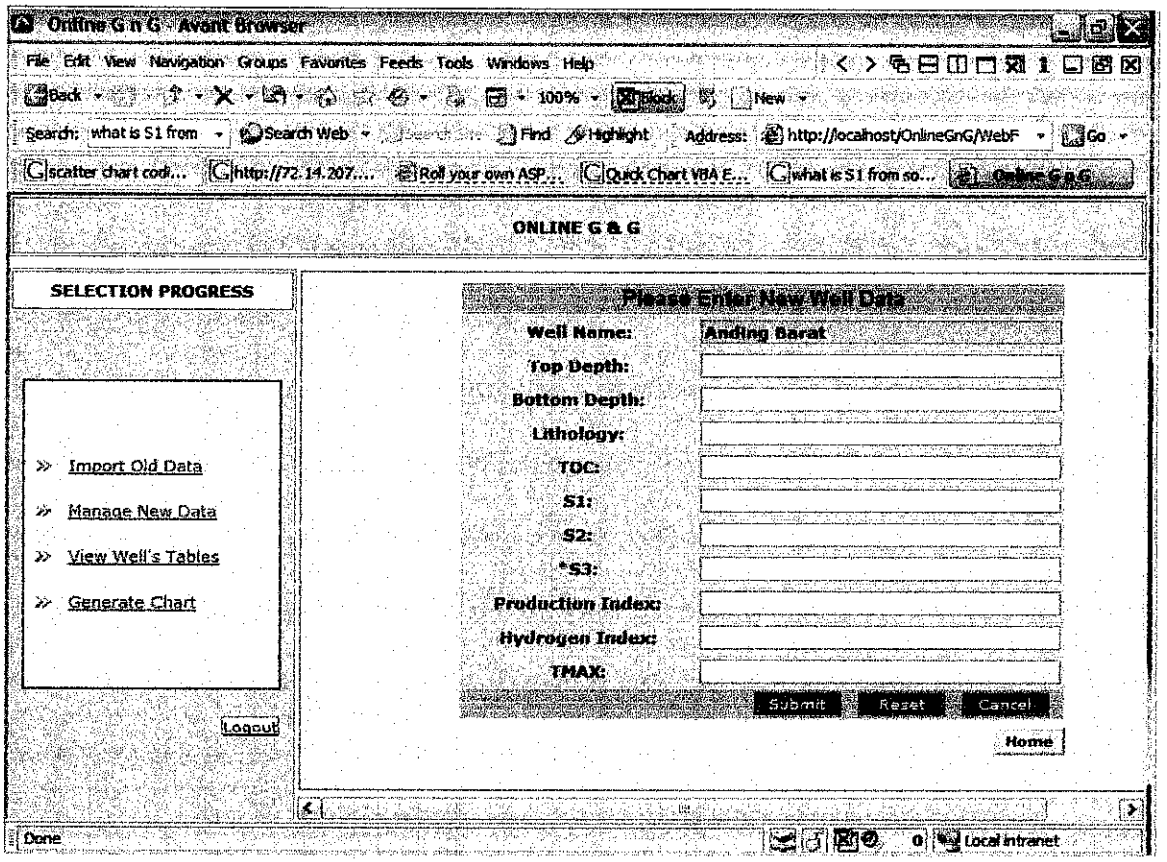


Figure 4.2.3.2.12: Second data entry for oil well's detail information

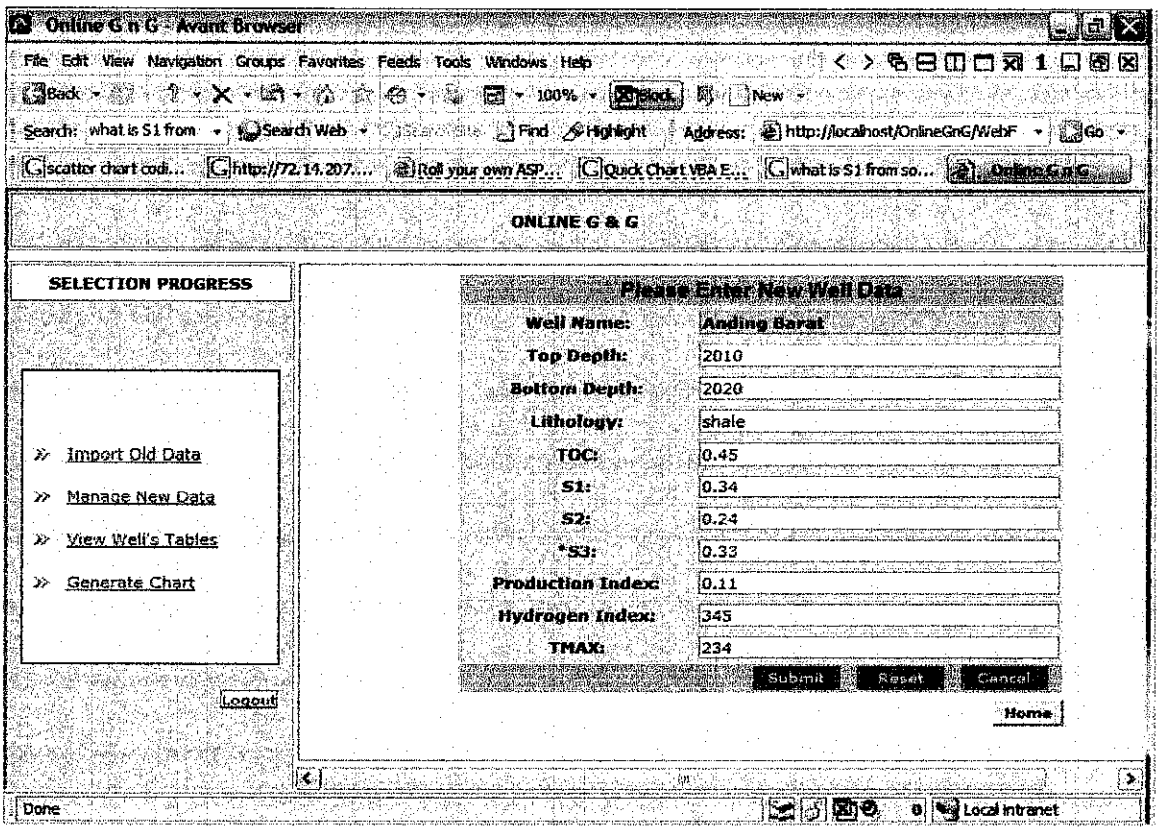


Figure 4.2.3.2.13: Enter information in each fields

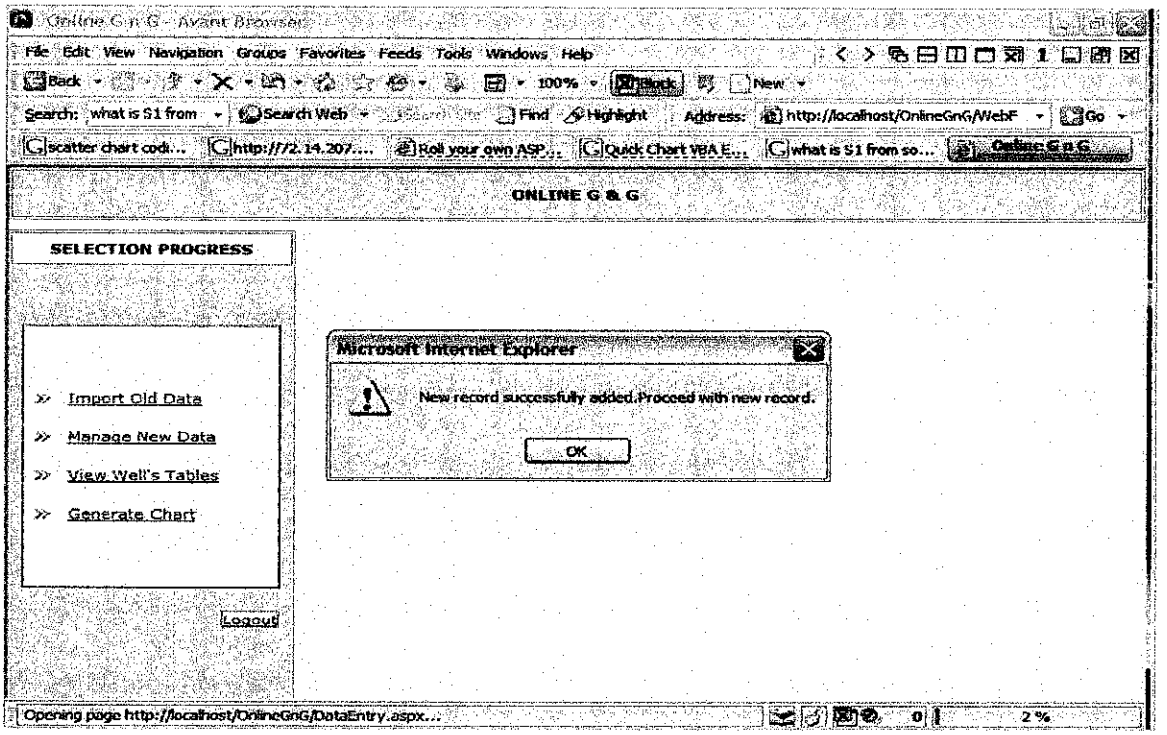


Figure 4.2.3.2.14: Pop-up message indicating successful record entry

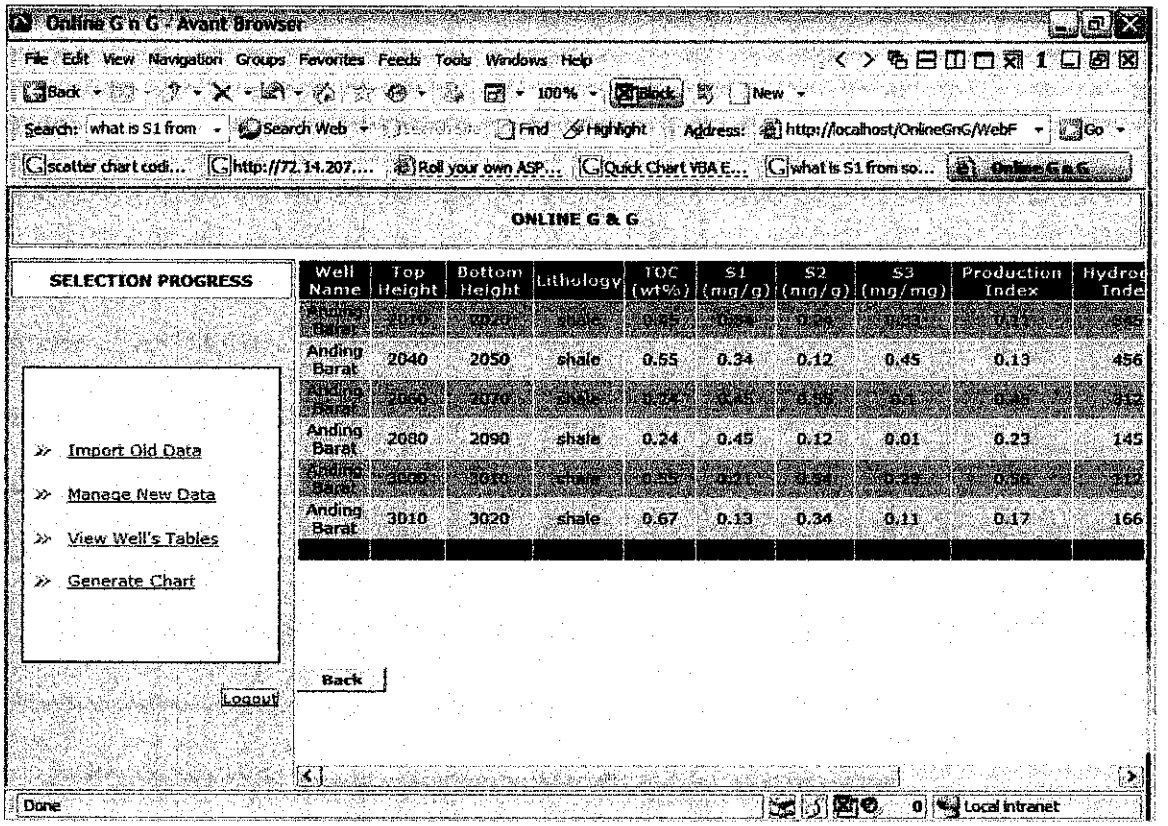


Figure 4.2.3.2.15: View oil well's detail information data table

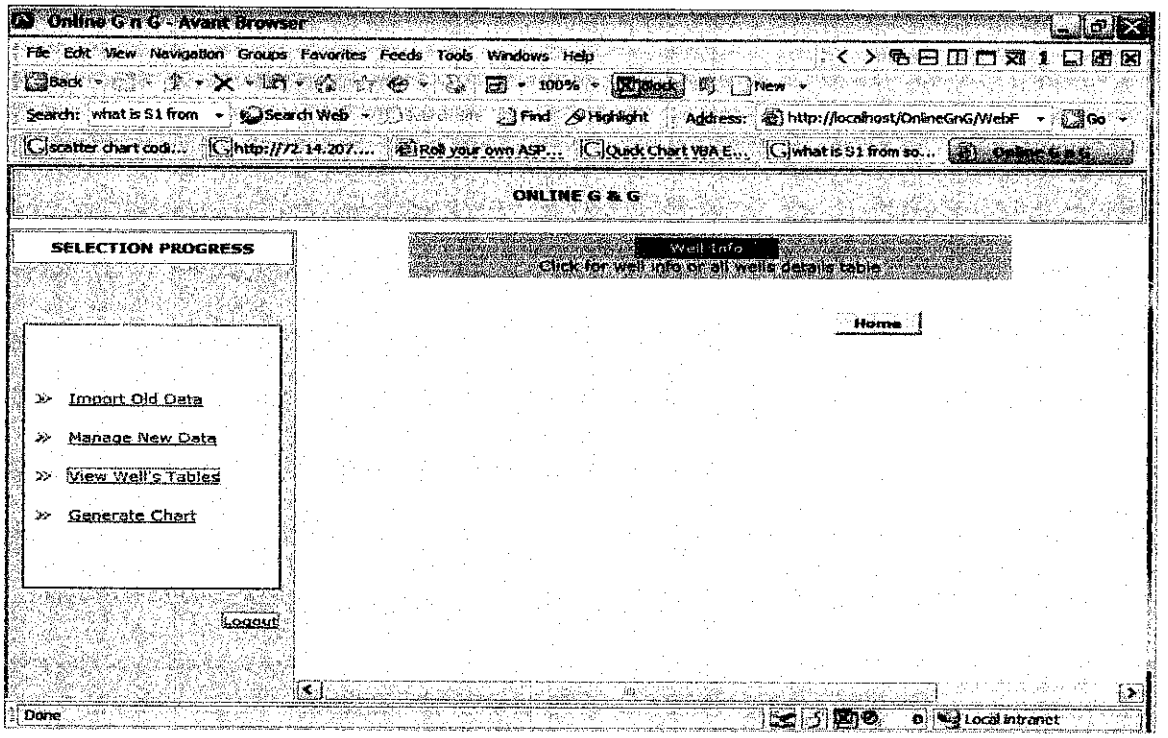


Figure 4.2.3.2.16: View general oil well data table page

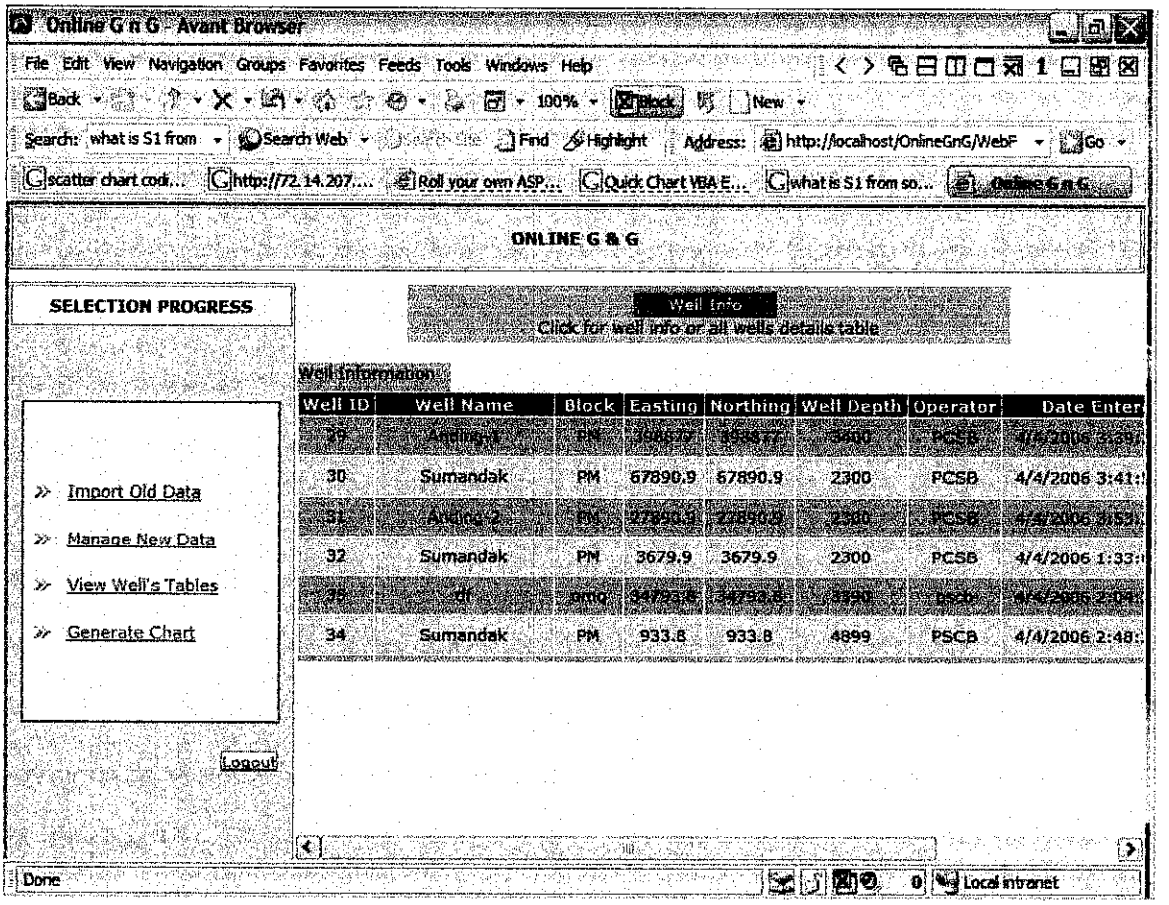


Figure 4.2.3.2.17: View general oil well data table

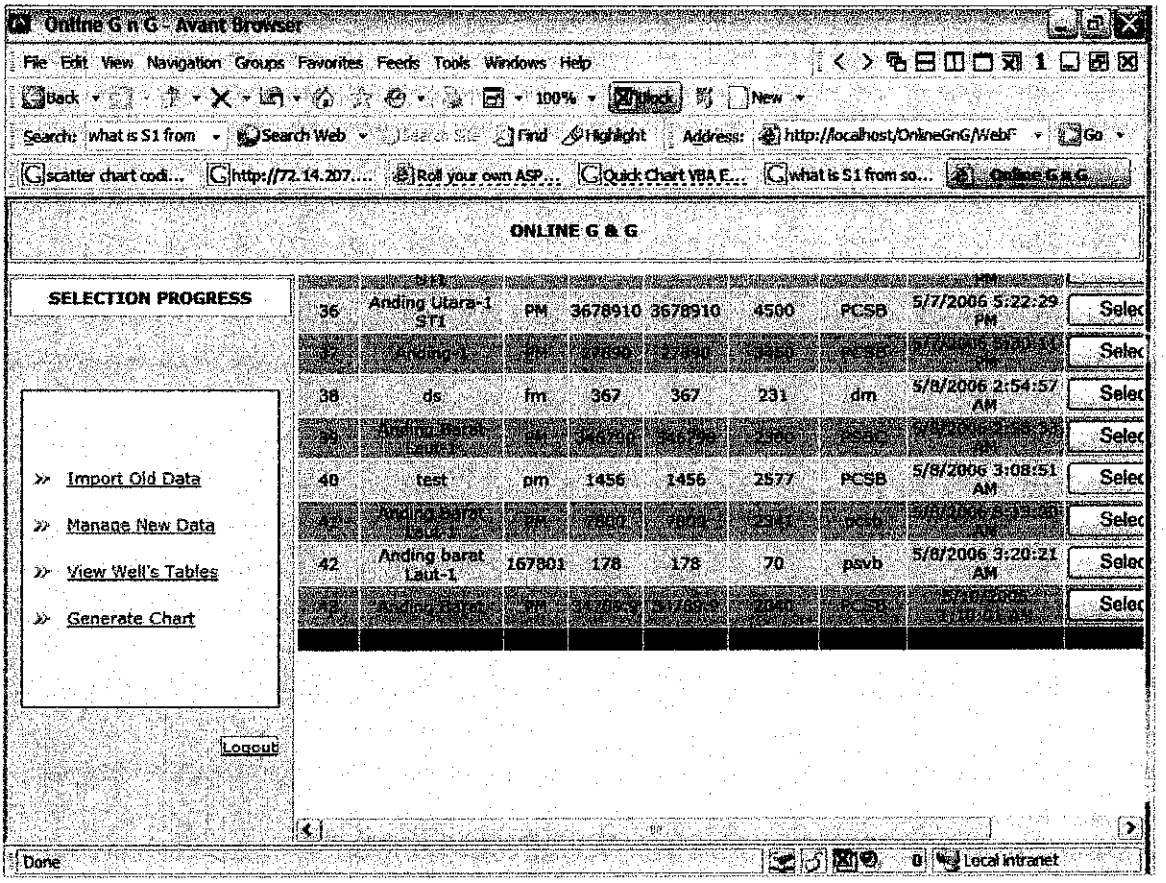


Figure 4.2.3.2.18: Button option in view general oil well data table for generating chart's procedure

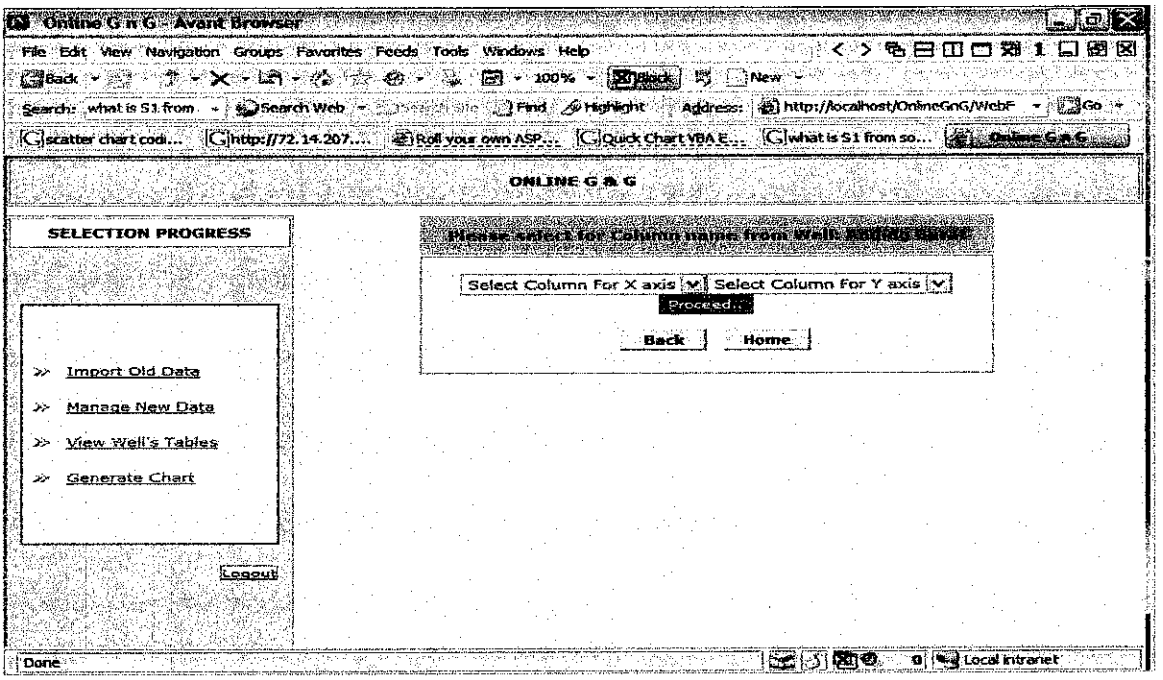


Figure 4.2.3.2.19: Select for X-axis and Y-axis values

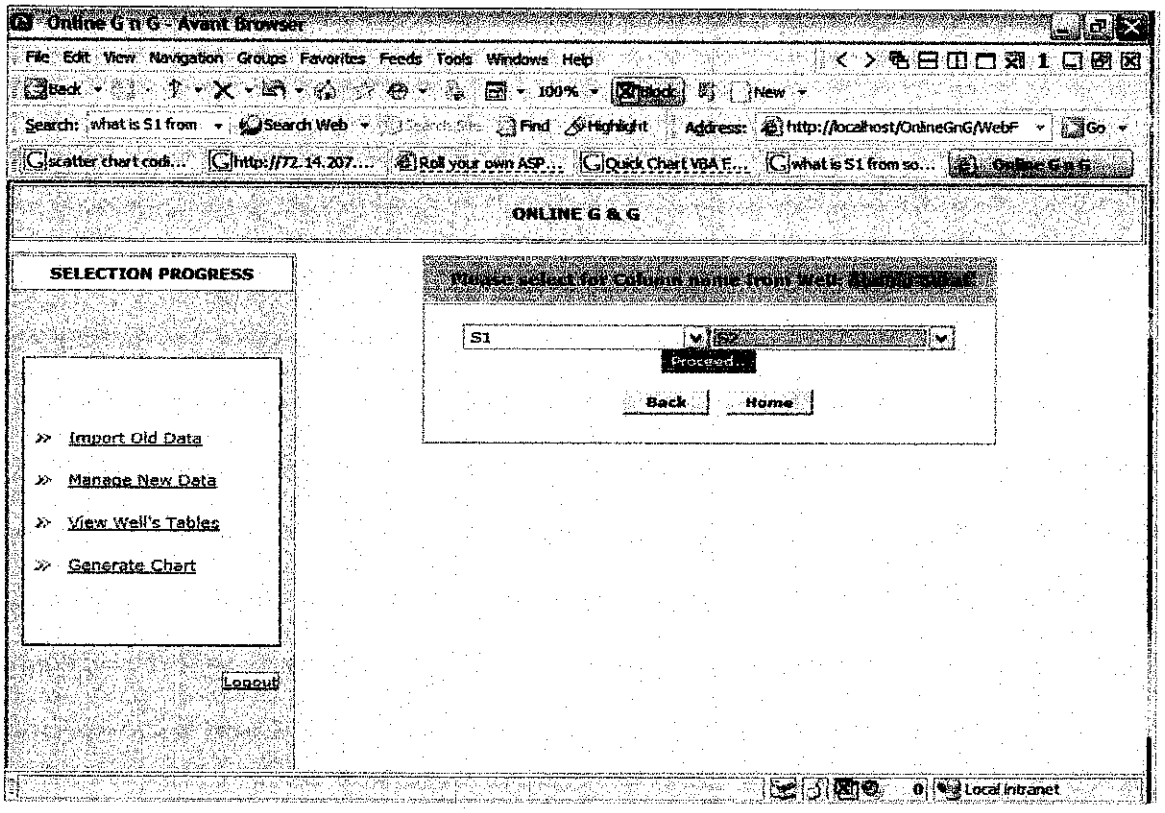


Figure 4.2.3.2.20: Selected attributes for X-axis and Y-axis values

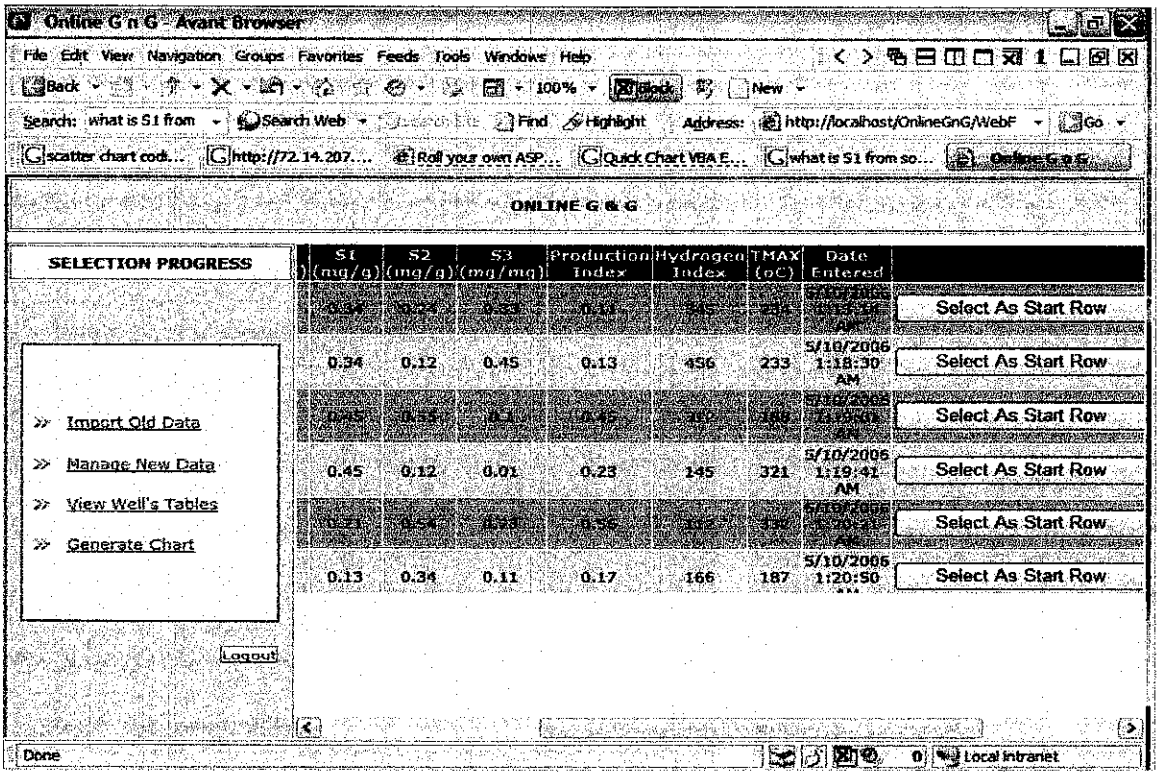


Figure 4.2.3.2.21: Continue with selecting start row for selected attribute

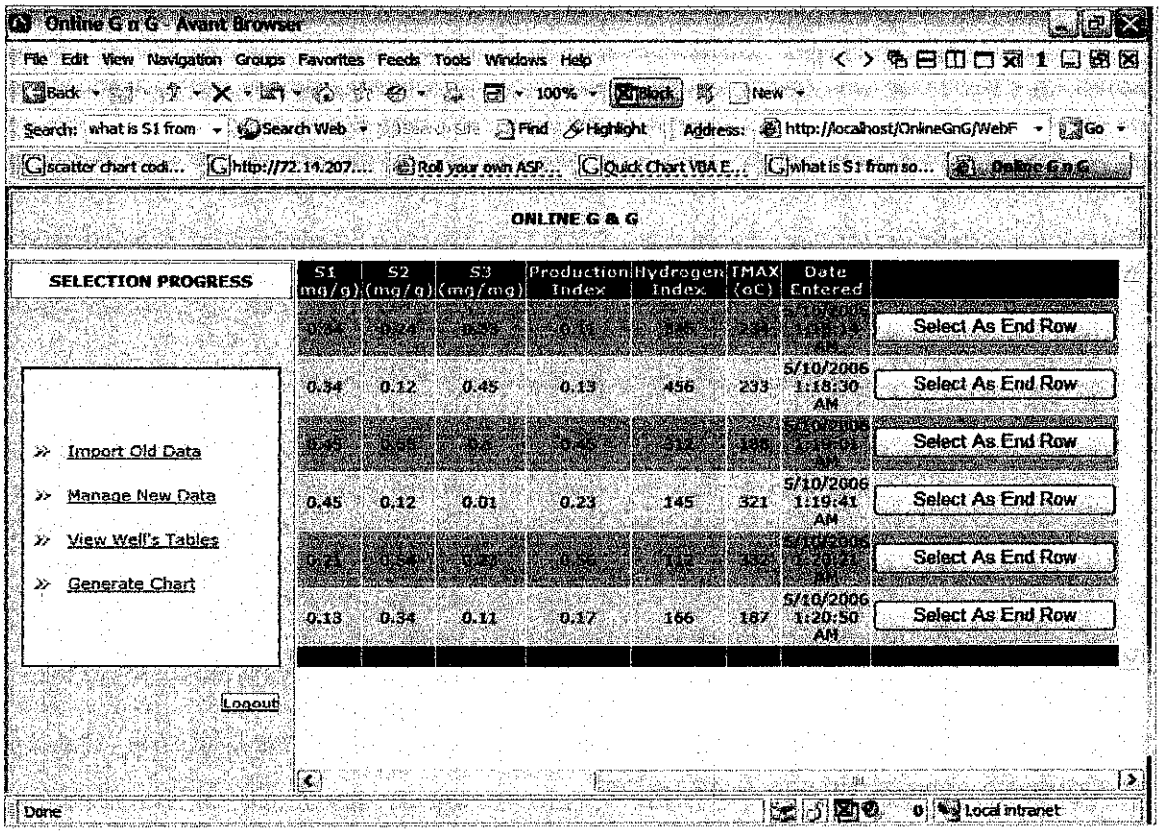


Figure 4.2.3.2.22: Continue with selecting end row for selected attribute

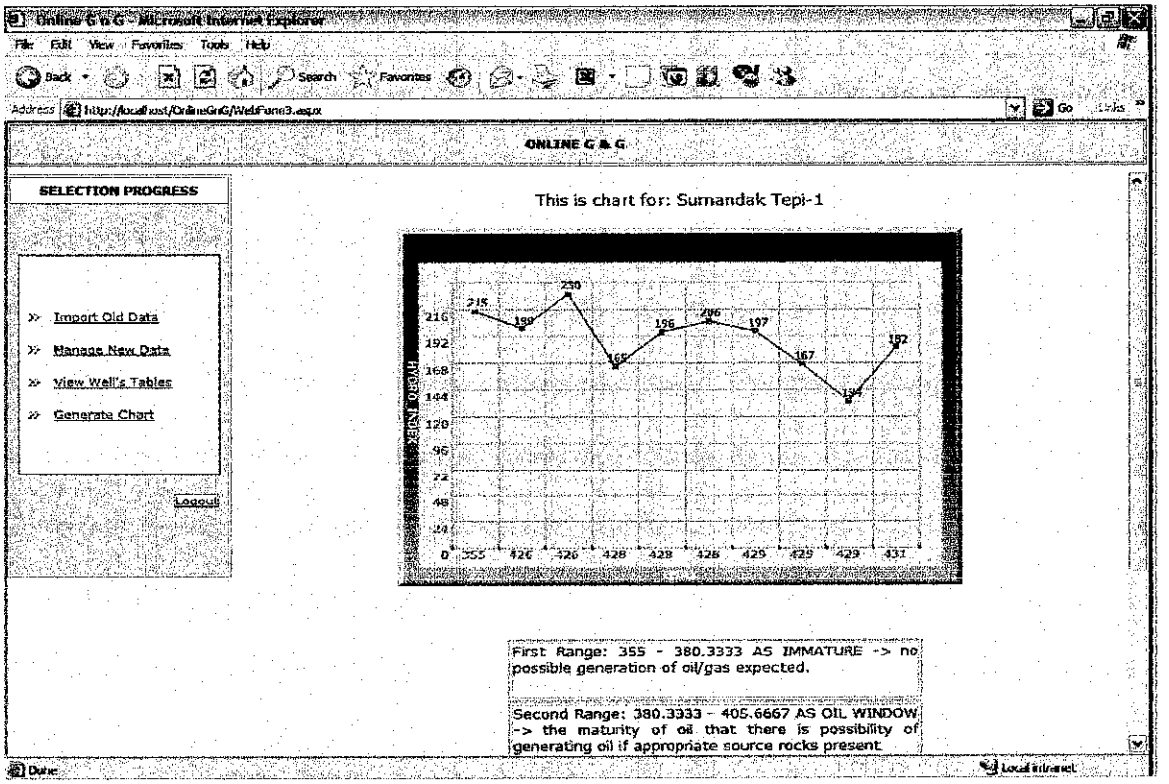


Figure 4.2.3.2.23: Generated chart from the selected attributes

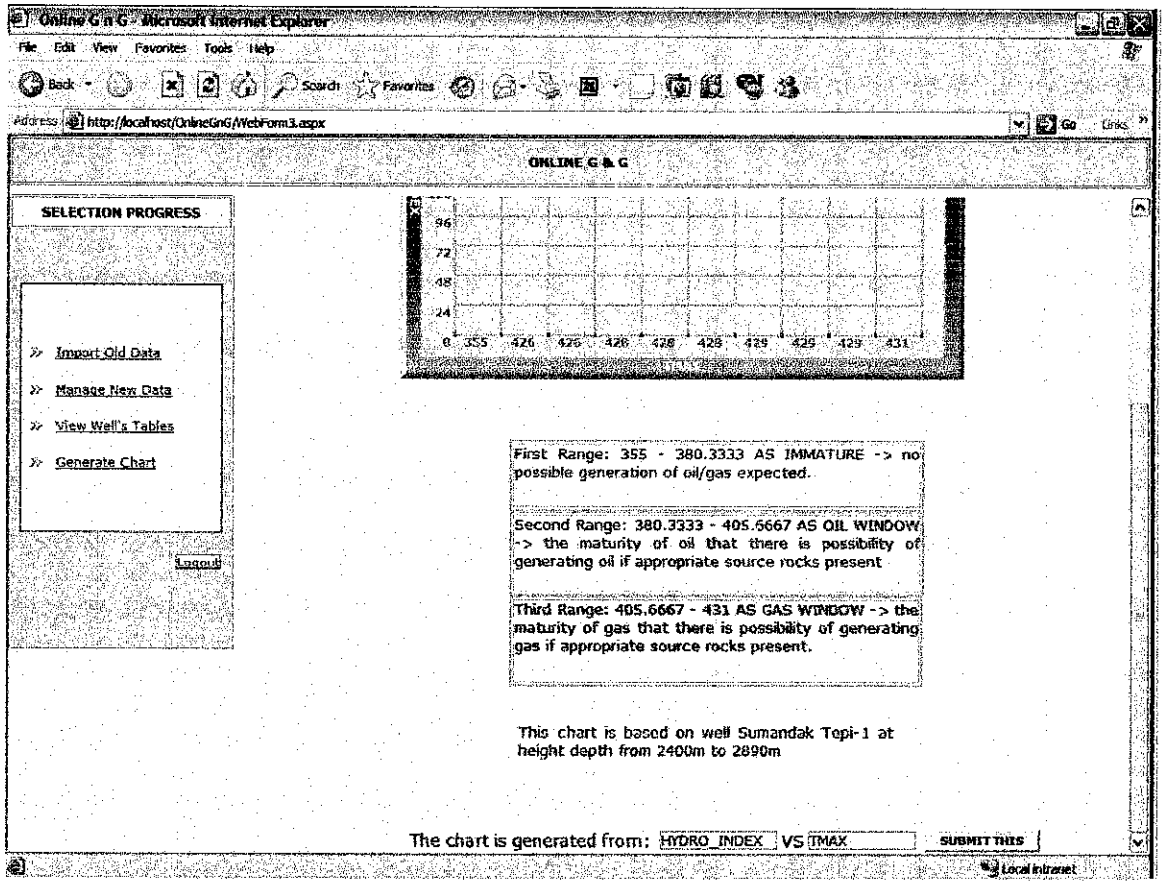


Figure 4.2.3.2.24: The yellow area is the reading for data interpretation

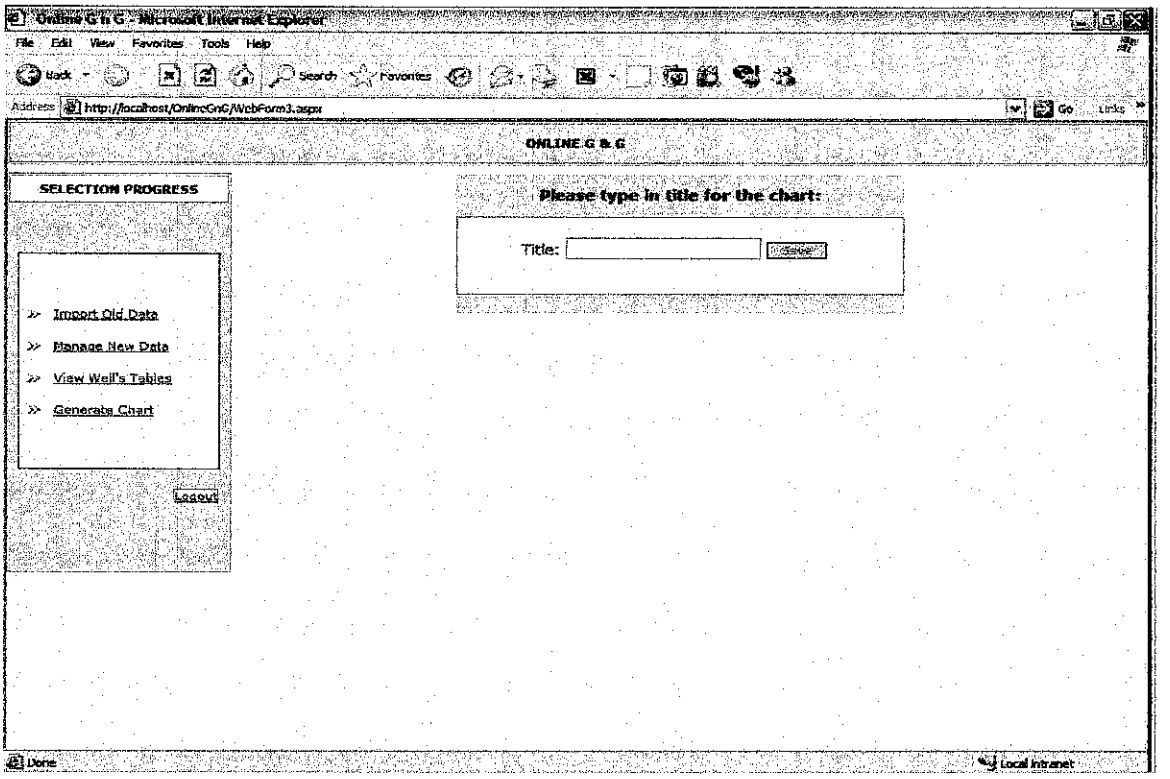


Figure 4.2.3.2.25: The submit button for title of the generated chart

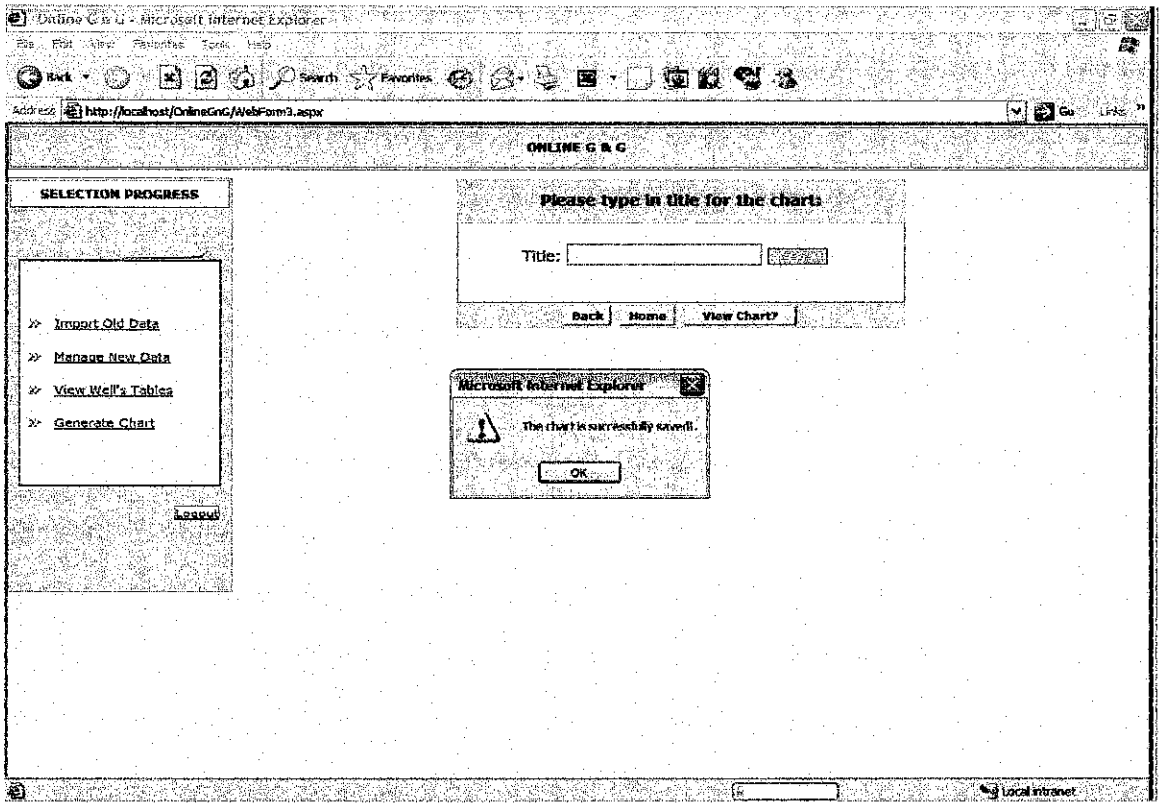


Figure 4.2.3.2.26: The submit button for title of the generated chart

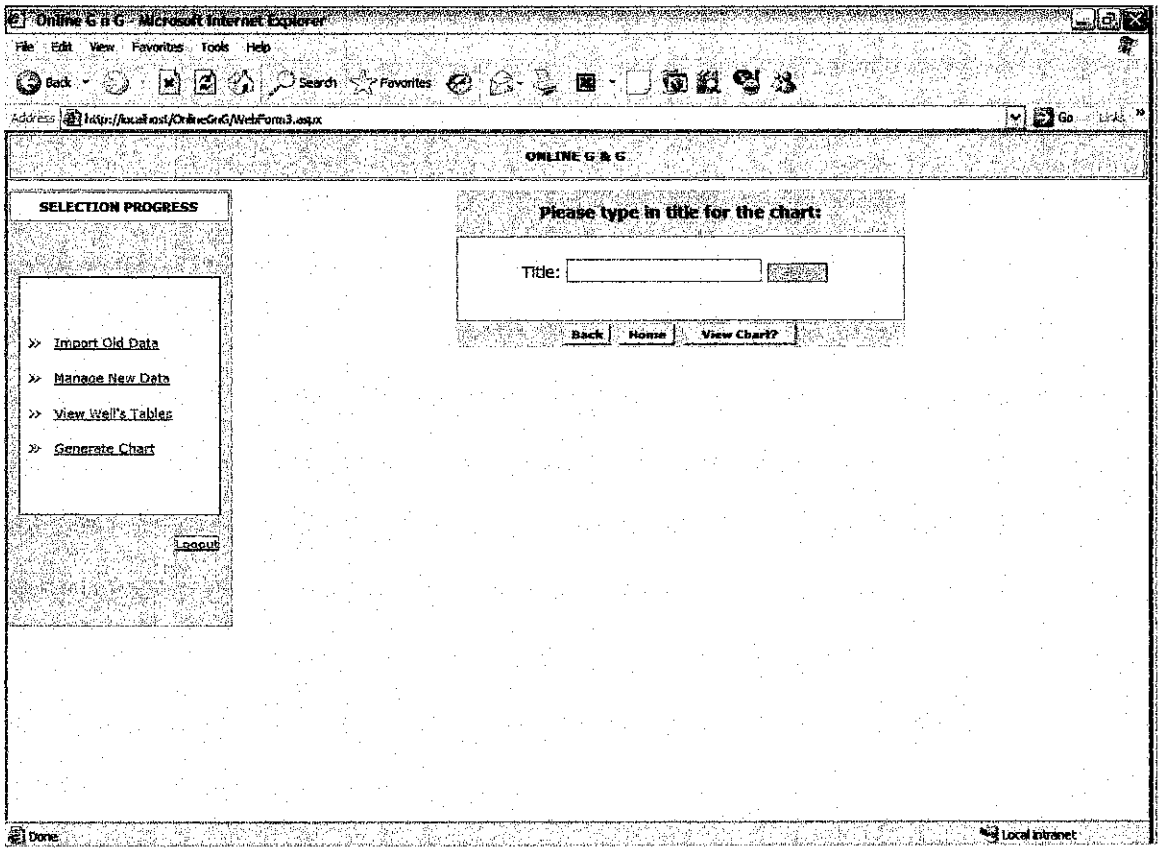


Figure 4.2.3.2.27: Option for user to select the chart being saved

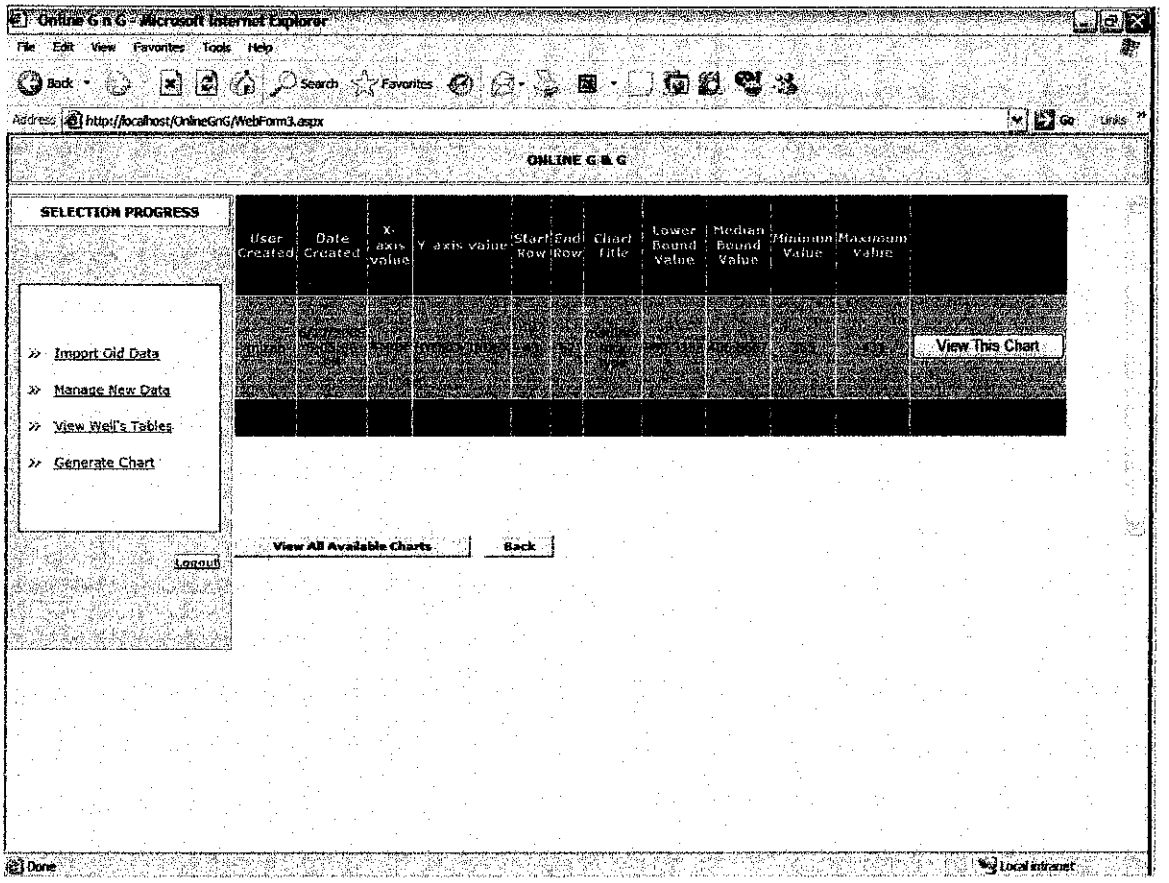


Figure 4.2.3.2.28: The table contains the data of the saved chart

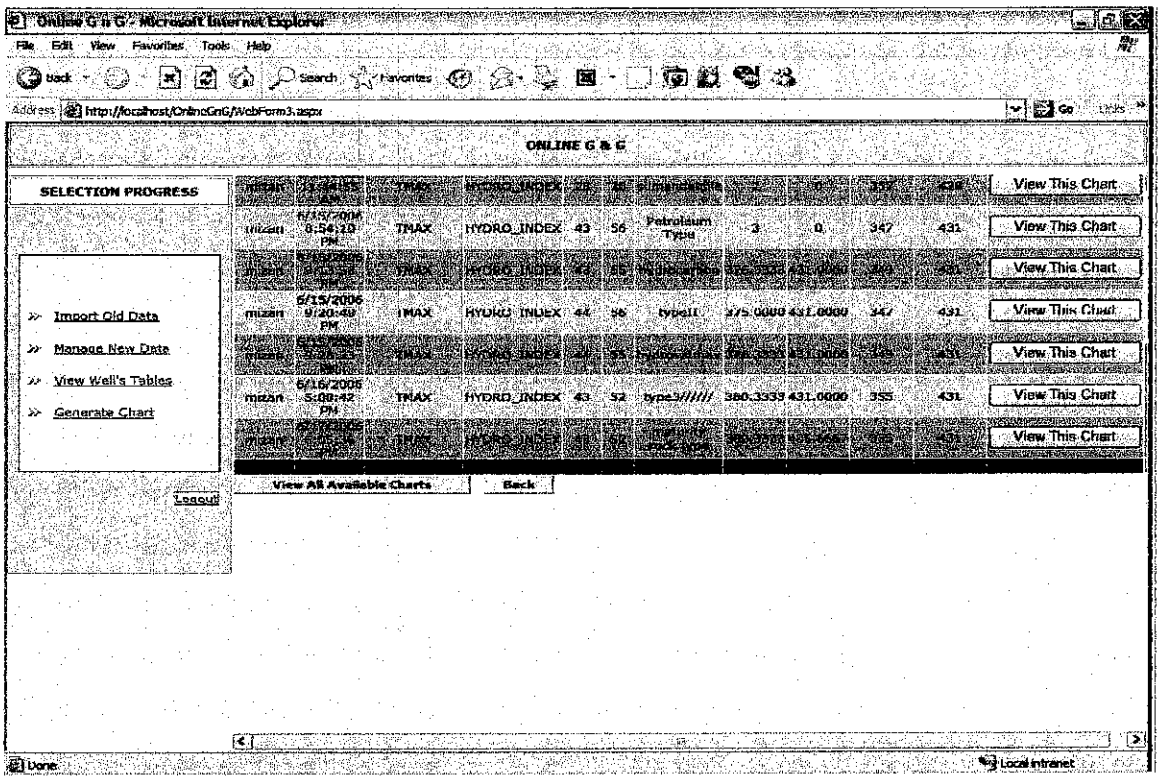


Figure 4.2.3.2.29: Option for viewing available chart that have been saved

```
Imports System.Data.SqlClient
Imports System.Web.security
Imports System.Data

Public Class login
    Inherits System.Web.UI.Page

// THIS IS A CODING FOR LOGIN PAGE

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.comm = New System.Data.SqlClient.SqlDataAdapter
        Me.SqlInsertCommand1 = New System.Data.SqlClient.SqlCommand
        Me.SqlSelectCommand1 = New System.Data.SqlClient.SqlCommand
        Me.dbconn = New System.Data.SqlClient.SqlConnection
        Me.dbset = New System.Data.DataSet
        Me.cmdGroups = New System.Data.SqlClient.SqlCommand
        CType(Me.dbset, System.ComponentModel.ISupportInitialize).BeginInit()
        '
        'comm
        '
        Me.comm.InsertCommand = Me.SqlInsertCommand1
        Me.comm.SelectCommand = Me.SqlSelectCommand1
        '
        'dbset
        '
        Me.dbset.DataSetName = "NewDataSet"
        Me.dbset.Locale = New System.Globalization.CultureInfo("en-US")
        '
        'cmdGroups
        '
        Me.cmdGroups.Connection = Me.dbconn
        CType(Me.dbset, System.ComponentModel.ISupportInitialize).EndInit()

    End Sub
    Protected WithEvents userid As System.Web.UI.WebControls.TextBox
    Protected WithEvents pass As System.Web.UI.WebControls.TextBox
    Protected WithEvents cmdLogin As System.Web.UI.WebControls.Button
    Protected WithEvents comm As System.Data.SqlClient.SqlDataAdapter
    Protected WithEvents SqlSelectCommand1 As System.Data.SqlClient.SqlCommand
    Protected WithEvents SqlInsertCommand1 As System.Data.SqlClient.SqlCommand
    Protected WithEvents dbconn As System.Data.SqlClient.SqlConnection
    Protected WithEvents dbset As System.Data.DataSet
    Protected WithEvents cmdGroups As System.Data.SqlClient.SqlCommand
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label

    'NOTE: The following placeholder declaration is required by the Web Form Designer.
    'Do not delete or move it.
    Private designerPlaceholderDeclaration As System.Object

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'Put user code to initialize the page here

        If Not IsPostBack Then

            Session("WhenEntered") = Now()

        End If
        Session("strconn") = "user id=sa;data source=MIMI;persist security info=True;initial catalog=GEODATA;password=sa"
```

```
End Sub

    Private Sub cmdLogin_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdLogin.Click


        Dim uid As String
        uid = Trim(userid.Text)
        uid = Replace(uid, "'", "")
        Dim passw As String
        passw = Trim(pass.Text)
        passw = Replace(passw, "'", "")

        Dim dbconn As New SqlConnection(Session("strconn"))
        Dim dbset As DataSet
        Dim comm As SqlDataAdapter

        dbconn.Open()


        Dim sql As String = " Select USER_ID From LOGIN_TB WHERE ((USER_FULLNAME = '" & userid.Text.Trim &
    "')and (USER_PASSWORD ='" & pass.Text.Trim & "'))"
        comm = New SqlDataAdapter(sql, dbconn)
        dbset = New DataSet
        comm.Fill(dbset, "login")

        If dbset.Tables("login").Rows.Count = 0 Then
          Label1.Text = "User not found. Please try other"
        Else
          Session("USER_ID") = dbset.Tables("login").Rows(0).Item("USER_ID")
          Session("USER_FULLNAME") = userid.Text

          PageRedirect()
          UpdateLogin()

        End If


    End Sub

    Sub PageRedirect()

Response.Redirect("WebForm3.aspx")


  End Sub

  Sub UpdateLogin()
    Dim dbconn As New SqlConnection(Session("strconn"))
    dbconn.Open()

    Dim cmdGroups As New SqlCommand("SP_UPDATE_LOGIN", dbconn)

    cmdGroups.CommandType = CommandType.StoredProcedure
    cmdGroups.Parameters.Add("@user_fullname", SqlDbType.VarChar, 50).Value = Session("USER_FULLNAME")
    cmdGroups.Parameters.Add("@login_date", SqlDbType.DateTime).Value = Now()

    Dim cmdRun As New SqlDataAdapter
    cmdRun.InsertCommand = cmdGroups
    cmdRun.InsertCommand.ExecuteNonQuery()

    dbconn.Close()

  End Sub

  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Response.Redirect("Profile.aspx")
  End Sub
End Class
```

// CODING FOR DATA ENTRY

```
Imports System.Data
Imports System.Data.SqlClient
Imports System.Data.OleDb

Public Class Import
   Inherits System.Web.UI.Page
   Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"


Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
      'CODEGEN: This method call is required by the Web Form Designer
      'Do not modify it using the code editor.
      InitializeComponent()
   End Sub

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
      'Put user code to initialize the page here
      If Not Page.IsPostBack Then

         Label1.Visible = False
         DataGrid1.Visible = False
         Button1.Visible = False
         Button2.Visible = False

         ddlRange.Items.Add("SourceRock")
         ddlRange.Items.Add("Sumandak")
      End If
   End Sub


Private Sub cmdCreate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCreate.Click

            Dim strmessage As String


      Session("range2") = ddlRange.SelectedItem.Text
      Session("range3") = ddlRange.SelectedItem.Text

      Label1.Visible = True
      Button2.Visible = True
      Button1.Visible = True

      strmessage = "Range Selected!"

Dim sConnectionString As String = "Provider=Microsoft.Jet.OLEDB.4.0;" _
         & "Data Source=" & Server.MapPath("../ExcelData.xls") _
         & ";" & "Extended Properties=Excel 8.0;"

      Dim objConn As New OleDbConnection(sConnectionString)

      ' Open connection with the database.

      Try
         objConn.Open()

      ' Create new OleDbCommand to return data from worksheet.

Dim objCmdSelect1 As New OleDbCommand("SELECT * FROM " & Session("range2"), objConn)
Dim objCmdSelect As New OleDbCommand("SELECT * FROM " & Session("range3"), objConn)

' Create new OleDbDataAdapter that is used to build a DataSet
' based on the preceding SQL SELECT statement.
Dim objAdapter1 As New OleDbDataAdapter
Dim objAdapter2 As New OleDbDataAdapter
```

```
' Pass the Select command to the adapter.
    objAdapter1.SelectCommand = objCmdSelect
    objAdapter2.SelectCommand = objCmdSelect1

' Create new DataSet to hold information from the worksheet.

Dim objDataset1 As New DataSet
Dim objDataset2 As New DataSet

' Fill the DataSet with the information from the worksheet.

                    objAdapter1.Fill(objDataset1, "XLData")
      objAdapter2.Fill(objDataset2, "XLData")

      Dim xldt As New DataTable
      objAdapter1.Fill(xldt)
      objAdapter1.Dispose()

      objAdapter2.Fill(xldt)
      objAdapter2.Dispose()


Dim conn As New SqlConnection("workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security
info=True;initial catalog=GEODATA;password=sa")

conn.Open()
Dim da As New SqlDataAdapter("select * from EXCEL_FILE", conn)
Dim cb As SqlCommandBuilder = New SqlCommandBuilder(da)
Dim ds As New DataSet


      da.Fill(ds, "EXCEL_FILE")
      For Each xlrow As DataRow In xldt.Rows

          Dim dr As DataRow = ds.Tables("EXCEL_FILE").NewRow

          For Each col As DataColumn In ds.Tables("EXCEL_FILE").Columns

          dr(col.ColumnName) = xlrow(col.ColumnName)
        Next
        ds.Tables("EXCEL_FILE").Rows.Add(dr)
          Next
      da.Update(ds, "EXCEL_FILE")
      da.Dispose()
      conn.Close()
      ' Clean up objects.
      objConn.Close()
    Catch ex As Exception
      'Return
    End Try

    Dim strScript As String = "<script language=JavaScript>"
    strScript += "alert(""" & strMessage & """);"
    strScript += "</script>"

    If (Not Page.IsStartupScriptRegistered("clientScript")) Then
      Page.RegisterStartupScript("clientScript", strScript)
    End If
  End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Response.Redirect("Welcome.aspx")

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

    Dim dbconn As New SqlConnection(strconn)
```

```
    Dim dbset As DataSet
    Dim comm As SqlDataAdapter
    DataGrid1.Visible = True

    Const strSql As String = "SELECT * FROM EXCEL_FILE"
    Dim objCmd As New SqlCommand(strSql, dbconn)
    dbconn.Open()
    DataGrid1.DataSource = objCmd.ExecuteReader(CommandBehavior.CloseConnection)
    DataGrid1.DataBind()
    dbconn.Close()

  End Sub
End Class


//CODING FOR MANAGE WELL

Imports System.Data
Imports System.Data.SqlClient


Public Class ManageWell
  Inherits System.Web.UI.Page

Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    InitializeComponent()
  End Sub

Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;datasource=MIMI;persist
securityinfo=True;initialcatalog=GEODATA;password=sa"

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here
    If Not IsPostBack Then
      Session("WhenEntered") = Now()
      cmdviewdata.Visible = False
    End If

  End Sub


Private Sub submission_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles submission.Click

    Dim strMessage As String

    If submission.Text = "Submit" Then
      If WellNameBaru.Text = "" Then
        strMessage = " Please enter data!"
      Else
        submitData()

      End If

    End If

    Dim strScript As String = "<script language=JavaScript>"
    strScript += "alert(""" & strMessage & """);"
    strScript += "</script>"

    If (Not Page.IsStartupScriptRegistered("clientScript")) Then
      Page.RegisterStartupScript("clientScript", strScript)
    End If
End Sub

Sub submitData()

Dim strMessage As String
```

```
        insertnew()
        strMessage = "New record successfully added. Click View Data button to view and add each well's record."
        cmdviewdata.Visible = True

    Dim strScript As String = "<script language=JavaScript>"
    strScript += "alert(""" & strMessage & """);"
    strScript += "</script>"
    If (Not Page.IsStartupScriptRegistered("clientScript")) Then
      Page.RegisterStartupScript("clientScript", strScript)
    End If

End Sub


Sub insertnew()
    Dim strMessage As String
    If WellNameBaru.Text <> "" Then
        Dim dbconn As New SqlConnection(strconn)
    dbconn.Open()

        Dim cmdNew As New SqlCommand("SP_GEO_INSERT_WELLSID", dbconn)
      cmdNew.CommandType = CommandType.StoredProcedure

      cmdNew.Parameters.Add("@well_name", SqlDbType.VarChar, 50).Value = WellNameBaru.Text & ""
      cmdNew.Parameters.Add("@block", SqlDbType.VarChar, 50).Value = block.Text & ""
      cmdNew.Parameters.Add("@easting", SqlDbType.Float).Value = easting.Text
      cmdNew.Parameters.Add("@northing", SqlDbType.Float).Value = northing.Text
      cmdNew.Parameters.Add("@well_depth", SqlDbType.Int).Value = CInt(well_depth.Text)
      cmdNew.Parameters.Add("@operator", SqlDbType.VarChar, 50).Value = operator.Text & ""
      cmdNew.Parameters.Add("@data_date", SqlDbType.DateTime).Value = Now(

            Dim cmdRun As New SqlDataAdapter
      cmdRun.InsertCommand = cmdNew
      cmdRun.InsertCommand.ExecuteNonQuery()

      dbconn.Close()

    End If

    strMessage = "New record successfully added."

    Dim strScript As String = "<script language=JavaScript>"
    strScript += "alert(""" & strMessage & """);"
    strScript += "</script>"

    If (Not Page.IsStartupScriptRegistered("clientScript")) Then
      Page.RegisterStartupScript("clientScript", strScript)
    End If

    Clear()
End Sub

Private Sub BtnReset_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnReset.Click

    Clear()
End Sub


Sub Clear()
    WellNameBaru.Text = ""
    block.Text = ""
    easting.Text = ""
    northing.Text = ""
    well_depth.Text = ""
    operator.Text = ""
End Sub

Private Sub cmdviewdata_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdviewdata.Click

    Response.Redirect("ViewNewWell.aspx")
```

```
    End Sub

Private Sub cancellation_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cancellation.Click

        Response.Redirect("ManageWell.aspx")
End Sub

Private Sub cmdHome_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdHome.Click
        Response.Redirect("Welcome.aspx")
End Sub
End Class




// CODING FOR DATA ENTRY


Imports System.Data.SqlClient
Imports System.Data


Public Class DataEntry
    Inherits System.Web.UI.Page
    Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"


Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
End Sub

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'Put user code to initialize the page here

        If Not IsPostBack Then
            Session("WhenEntered") = Now()
            cmdviewdata.Visible = False
            WellNameBaru.Text = Session("WELLNAME")
            WellNameBaru.ReadOnly = True
        End If
End Sub

Private Sub submitbutton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles submitbutton.Click

        Dim strMessage As String

        If submitbutton.Text = "Submit" Then
            If top_depth.Text = "" Then
                strMessage = " Please enter data!"
            Else
                submitData()
            End If
        End If

        Dim strScript As String = "<script language=JavaScript>"
        strScript += "alert(""" & strMessage & """);"
        strScript += "</script>"

        If (Not Page.IsStartupScriptRegistered("clientScript")) Then
            Page.RegisterStartupScript("clientScript", strScript)
        End If
    End Sub

Sub submitData()

    Dim strMessage As String
```

```
        insertnew()
    strMessage = "New record successfully added.Click View Data button to view record."
    cmdviewdata.Visible = True

    Dim strScript As String = "<script language=JavaScript>"
    strScript += "alert(""" & strMessage & """);"
    strScript += "</script>"

    If (Not Page.IsStartupScriptRegistered("clientScript")) Then
      Page.RegisterStartupScript("clientScript", strScript)
    End If

End Sub


Sub insertnew()

Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"

        Dim strMessage As String

    If top_depth.Text <> "" Then

      Dim dbconn As New SqlConnection(strconn)
      dbconn.Open()

    Dim cmdNew As New SqlCommand("SP_GEO_INSERT_RECNUM", dbconn)
      cmdNew.CommandType = CommandType.StoredProcedure

      cmdNew.Parameters.Add("@well_name", SqlDbType.VarChar, 50).Value = WellNameBaru.Text
      cmdNew.Parameters.Add("@top_height", SqlDbType.Int).Value = CInt(top_depth.Text)
      cmdNew.Parameters.Add("@bottom_height", SqlDbType.Int).Value = CInt(bottom_depth.Text)
      cmdNew.Parameters.Add("@lithology", SqlDbType.VarChar, 50).Value = lithology.Text
      cmdNew.Parameters.Add("@toc", SqlDbType.Float).Value = toc.Text
      cmdNew.Parameters.Add("@s1", SqlDbType.Float).Value = s1.Text
      cmdNew.Parameters.Add("@s2", SqlDbType.Float).Value = s2.Text
      cmdNew.Parameters.Add("@tmax", SqlDbType.Float).Value = tmax.Text
      cmdNew.Parameters.Add("@s3", SqlDbType.Float).Value = s3.Text
      cmdNew.Parameters.Add("@prod_index", SqlDbType.Float).Value = prod_index.Text
      cmdNew.Parameters.Add("@hydro_index", SqlDbType.Int).Value = CInt(hydro_index.Text)
      cmdNew.Parameters.Add("@data_date", SqlDbType.DateTime).Value = Now()


      Dim cmdRun As New SqlDataAdapter
      cmdRun.InsertCommand = cmdNew
      cmdRun.InsertCommand.ExecuteNonQuery()

      dbconn.Close()

    End If
    strMessage = "New record successfully added.Proceed with new record."

    Dim strScript As String = "<script language=JavaScript>"
    strScript += "alert(""" & strMessage & """);"
    strScript += "</script>"

    If (Not Page.IsStartupScriptRegistered("clientScript")) Then
      Page.RegisterStartupScript("clientScript", strScript)
    End If

    Clear()

End Sub

Private Sub cmdReset_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdReset.Click

    bottom_depth.Text = ""
    top_depth.Text = ""
    lithology.Text = ""
```

```
        toc.Text = ""
        s1.Text = ""
        s2.Text = ""
        tmax.Text = ""
        s3.Text = ""
        prod_index.Text = ""
        hydro_index.Text = ""

End Sub

    Sub Clear()

        bottom_depth.Text = ""
        top_depth.Text = ""
        lithology.Text = ""
        toc.Text = ""
        s1.Text = ""
        s2.Text = ""
        tmax.Text = ""
        s3.Text = ""
        prod_index.Text = ""
        hydro_index.Text = ""
    End Sub

Private Sub cmdviewdata_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdviewdata.Click

        Response.Redirect("ViewDataGrid.aspx")
End Sub

Private Sub cmdCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCancel.Click
        Response.Redirect("DataEntry.aspx")
    End Sub

Private Sub cmdButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdButton.Click
        Response.Redirect("Managewell.aspx")
    End Sub
End Class


Imports System.Data
Imports System.Data.SqlClient



// CODING FOR VIEWING GENERAL OIL WELL INFORMATION
Public Class ViewDataGrid
    Inherits System.Web.UI.Page

Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

                    If Not IsPostBack Then
        Dim dbconn As New SqlConnection(strconn)
        Dim dbset As DataSet
        Dim comm As SqlDataAdapter
        Dim strSql As String = "SELECT * FROM DATASTORE_TB WHERE WELL_NAME = '" & Session("WELLNAME") &
"'"
        Dim objCmd As New SqlCommand(strSql, dbconn)
        dbconn.Open()
                    welldatagrid.DataSource = objCmd.ExecuteReader(CommandBehavior.CloseConnection)
        welldatagrid.DataBind()
        dbconn.Close()
      End If
End Sub

Private Sub backbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles backbtn.Click
        Response.Redirect("DataEntry.aspx")
    End Sub
```

```
End Class


//CODING FOR VIEWING OIL WELL'S DETAIL INFORMATION

Imports System.Data
Imports System.Data.SqlClient


Public Class ViewDataGridOnly
  Inherits System.Web.UI.Page

Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    InitializeComponent()
End Sub
Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    If Not IsPostBack Then
      Dim dbconn As New SqlConnection(strconn)
      Dim dbset As DataSet
      Dim comm As SqlDataAdapter

      Const strSql As String = " SELECT * FROM DATASTORE_TB"
      Dim objCmd As New SqlCommand(strSql, dbconn)
      dbconn.Open()
      welldatagrid.DataSource = objCmd.ExecuteReader(CommandBehavior.CloseConnection)
      welldatagrid.DataBind()

      dbconn.Close()
    End If
End Sub

Private Sub backbtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles backbtn.Click
    Response.Redirect("Charting.aspx")
  End Sub
End Class



// CODING FOR CHOOSING WELL FOR GENERATING CHART

Imports System.Data
Imports System.Data.SqlClient

Public Class ChooseWell
  Inherits System.Web.UI.Page
Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    InitializeComponent()
  End Sub

Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

      If Not IsPostBack Then
    Dim dbconn As New SqlConnection(strconn)
    Dim dbset As DataSet
    Dim comm As SqlDataAdapter
    Const strSql As String = " SELECT * FROM WELLSINFO_TB"
    Dim objCmd As New SqlCommand(strSql, dbconn)
```

```
        dbconn.Open()
        NewWell.DataSource = objCmd.ExecuteReader(CommandBehavior.CloseConnection)
        NewWell.DataBind()
        dbconn.Close()
    End If

End Sub

Sub Select_ItemCommand(ByVal sender As Object, ByVal e As DataGridCommandEventArgs)

    Session("CHOOSEWELL") = e.Item.Cells(1).Text
    Response.Redirect("Chart.aspx")
End Sub
End Class


//CODING FOR SELECTING X AND Y VALUES

Imports System.Data
Imports System.Data.SqlClient
Imports WebChart


Public Class Chart
  Inherits System.Web.UI.Page
Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"


Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    InitializeComponent()
End Sub

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    If Not IsPostBack Then

        Label1.Text = Session("CHOOSEWELL")

        ddlX.Items.Add("TOC")
        ddlX.Items.Add("S1")
        ddlX.Items.Add("S2")
        ddlX.Items.Add("PROD_INDEX")
        ddlX.Items.Add("HYDRO_INDEX")
        ddlX.Items.Add("TMAX")
        ddlY.Items.Add("TOC")
        ddlY.Items.Add("S1")
        ddlY.Items.Add("S2")
        ddlY.Items.Add("PROD_INDEX")
        ddlY.Items.Add("HYDRO_INDEX")
        ddlY.Items.Add("TMAX")

    End If
End Sub

Private Sub cmdGenerate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdGenerate.Click

    Session("X") = ddlX.SelectedItem.Text
    Session("Y") = ddlY.SelectedItem.Text

    Response.Redirect("StartRow.aspx")
  End Sub
End Class
```

// CODING FOR SELECTING START ROW

```
Imports System.Data.SqlClient
Imports System.Data

Public Class StartRow
   Inherits System.Web.UI.Page
   Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"


Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init

    InitializeComponent()
End Sub

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here
    If Not IsPostBack Then
                Dim dbconn As New SqlConnection(strconn)
      Dim dbset As DataSet
      Dim comm As SqlDataAdapter
                Dim strSql As String = "SELECT * FROM DATASTORE_TB WHERE WELL_NAME = '" &
        Session("CHOOSEWELL") & "'"
      Dim objCmd As New SqlCommand(strSql, dbconn)
      dbconn.Open()
      StartGrid.DataSource = objCmd.ExecuteReader(CommandBehavior.CloseConnection)
      StartGrid.DataBind()
      dbconn.Close()
    End If
End Sub

Sub StartGrid_ItemCommand(ByVal sender As Object, ByVal e As DataGridCommandEventArgs)
    Session("StartRow") = e.Item.Cells(0).Text
    Response.Redirect("EndRow.aspx")
End Sub

Private Sub StartGrid_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
StartGrid.SelectedIndexChanged
End Sub
End Class
```

// CODING FOR SELECTING END ROW

```
Imports System.Data
Imports System.Data.SqlClient

Public Class EndRow
   Inherits System.Web.UI.Page
   Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"

Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
    InitializeComponent()
End Sub

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here

    If Not IsPostBack Th
      Dim dbconn As New SqlConnection(strconn)
      Dim dbset As DataSet
      Dim comm As SqlDataAdapter
```

```
                Dim strSql As String = "SELECT * FROM DATASTORE_TB WHERE WELL_NAME = '" &
        Session("CHOOSEWELL") & "' AND REC_NUM >= " & CInt(Session("StartRow"))

        Dim objCmd As New SqlCommand(strSql, dbconn)
        dbconn.Open()
        EndGrid.DataSource = objCmd.ExecuteReader(CommandBehavior.CloseConnection)
        EndGrid.DataBind()

        dbconn.Close()
    End If
End Sub

Sub EndGrid_ItemCommand(ByVal sender As Object, ByVal e As DataGridCommandEventArgs)

    ' Session("EndRow") = e.Item.Cells(0).Text
    Response.Redirect("FinalPage.aspx")

End Sub
End Class


//CODING FOR CHART AND DATA INTERPRETATION

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here
    If Not IsPostBack Then



        Label2.Text = Session("CHOOSEWELL")
        Label1.Text = "This is chart for:"

        ChartControl1.Visible = True

        Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
         catalog=GEODATA;password=sa"
        Dim reader As SqlDataReader

        Dim myConnection As New SqlConnection(strconn)


        myConnection.Open()

        ' to select for graph manipulation
        Dim strSQL As String = "SELECT " & Session("X") & " , " & Session("Y") & " FROM DATASTORE_TB WHERE
         REC_NUM >= " & CInt(Session("StartRow")) & " AND REC_NUM <= " & CInt(Session("EndRow")) & " ORDER BY "
         & Session("X") & " ASC"

        Dim myCommand = New SqlCommand(strSQL, myConnection)

        reader = myCommand.ExecuteReader()


        'chart object

        Dim chart As LineChart = New LineChart


        'bind data

        chart.DataSource = reader
        chart.DataXValueField = Session("X")
        chart.DataYValueField = Session("Y")


        ChartControl1.XTitle.Text = Session("X")
        ChartControl1.YTitle.Text = Session("Y")

        chart.ShowLegend = True
```

```
        chart.DataBind()
        chart.DataLabels.Visible = True

        'TO SHOW RESULT OF GRAPH AND INTERPRETATION
        'attach chart
        ChartControl1.Charts.Add(chart)
        ChartControl1.GridLines = GridLines.Both

        ChartControl1.RedrawChart()
        Label4.Text = Session("Y")
        Label6.Text = Session("X")

        Display()

        reader.Close()
        myConnection.Close()

    End If

End Sub

Function Display()

    Dim Total As Single
    Dim AmtEach As Single

    Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"

    Dim reader1 As SqlDataReader
    Dim reader2 As SqlDataReader


    Dim myConnection1 As New SqlConnection(strconn) 'this is for min_value
    Dim myConnection2 As New SqlConnection(strconn) ' this is for max_value


    myConnection1.Open()
    myConnection2.Open()


    'to define min and max value for both start and end row


    Dim strSQL1 As String = "SELECT MIN (" & Session("X") & ") AS MIN_COL FROM DATASTORE_TB WHERE
REC_NUM >= " & CInt(Session("StartRow")) & " AND REC_NUM <= " & CInt(Session("EndRow"))
    Dim strSQL2 As String = "SELECT MAX (" & Session("X") & ") AS MAX_COL FROM DATASTORE_TB WHERE
REC_NUM >= " & CInt(Session("StartRow")) & " AND REC_NUM <= " & CInt(Session("EndRow"))

    Dim myCommand1 = New SqlCommand(strSQL1, myConnection1)
    Dim myCommand2 = New SqlCommand(strSQL2, myConnection2)
    reader1 = myCommand1.ExecuteReader()
    reader2 = myCommand2.ExecuteReader()


    reader1.Read()
    reader2.Read()

    Session("min_value") = CSng(reader1("MIN_COL"))
    Session("max_value") = CSng(reader2("MAX_COL"))

    'to calculate each category
    Total = (Session("max_value")) - (Session("min_value"))

    AmtEach = (Total / 3)
    Dim LOW As Double = Session("min_value") + AmtEach
    Session("LOW") = LOW.ToString(Format("F4"))

    Dim MED As Double = Session("min_value") + AmtEach + AmtEach
```

```
        Session("MED") = MED.ToString(Format("F4"))


        ' to display

        LNo.Text = "First Range: " & Session("min_value") & " - " & Session("LOW") & " AS IMMATURE -> no possible
generation of oil/gas expected. "

        MNo.Text = "Second Range: " & Session("LOW") & " - " & Session("MED") & " AS OIL WINDOW -> the maturity of oil
that there is possibility of generating oil if appropriate source rocks present"
        'StatOil.Text = "OIL WINDOW"

        HNo.Text = "Third Range: " & Session("MED") & " - " & Session("max_value") & " AS GAS WINDOW -> the maturity of
gas that there is possibility of generating gas if appropriate source rocks present. "
        'StatGas.Text = "GAS WINDOW"

        lblInfo.Text = " This chart is based on well " & Session("CHOOSEWELL") & " at height depth from " & Session("HEIGHT")
& "m to " & Session("BOTTOM") & "m"


        reader1.Close()
        reader2.Close()

        myConnection1.Close()
        myConnection2.Close()


    End Function

    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
        Response.Redirect("saveChart.aspx")
    End Sub



'SAVING CHART FUNCTION

Imports System.Data
Imports System.Data.SqlClient

Public Class saveChart
    Inherits System.Web.UI.Page
    Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"


Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        If Not IsPostBack Then
            BackBtn.Visible = False
            HomeBtn.Visible = False
            viewBtn.Visible = False
        End If
    End Sub

    Private Sub saveBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles saveBtn.Click
Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"
        Dim strMessage As String


        Session("Title") = TextBox1.Text
        If saveBtn.Text = "Save" Then
            If TextBox1.Text = "" Then
                strMessage = " Please enter data!"
            Else
                SaveData()

            End If
```

```
    End If


  Dim strScript As String = "<script language=JavaScript>"
    strScript += "alert(""" & strMessage & """);"
    strScript += "</script>"

    If (Not Page.IsStartupScriptRegistered("clientScript")) Then
      Page.RegisterStartupScript("clientScript", strScript)
    End If
  End Sub


  Sub SaveData()
    Dim strMessage As String


    If TextBox1.Text <> "" Then

      Dim dbconn As New SqlConnection(strconn)
      dbconn.Open()

      Dim cmdNew As New SqlCommand("SP_GEO_INSERT_CHARTINFO", dbconn)
      cmdNew.CommandType = CommandType.StoredProcedure


      cmdNew.Parameters.Add("@user_login", SqlDbType.VarChar, 50).Value = Session("USER_FULLNAME")
      cmdNew.Parameters.Add("@date_time", SqlDbType.DateTime).Value = Now()
      cmdNew.Parameters.Add("@data_x", SqlDbType.VarChar, 50).Value = Session("X")
      cmdNew.Parameters.Add("@data_y", SqlDbType.VarChar, 50).Value = Session("Y")
      cmdNew.Parameters.Add("@start_row", SqlDbType.VarChar, 50).Value = Session("StartRow")
      cmdNew.Parameters.Add("@end_row", SqlDbType.VarChar, 50).Value = Session("EndRow")
      cmdNew.Parameters.Add("@title", SqlDbType.VarChar, 50).Value = Session("Title")
      cmdNew.Parameters.Add("@low_param", SqlDbType.VarChar, 50).Value = Session("LOW")
      cmdNew.Parameters.Add("@med_param", SqlDbType.VarChar, 50).Value = Session("MED")
      'cmdNew.Parameters.Add("@high_param", SqlDbType.VarChar, 50).Value = Session("HIGH")
      cmdNew.Parameters.Add("@min_value", SqlDbType.VarChar, 50).Value = Session("min_value")
      cmdNew.Parameters.Add("@max_value", SqlDbType.VarChar, 50).Value = Session("max_value")



      Dim cmdRun As New SqlDataAdapter
      cmdRun.InsertCommand = cmdNew
      cmdRun.InsertCommand.ExecuteNonQuery()

      dbconn.Close()

    End If

    strMessage = "The chart is successfully saved!."

    Dim strScript As String = "<script language=JavaScript>"
    strScript += "alert(""" & strMessage & """);"
    strScript += "</script>"

    If (Not Page.IsStartupScriptRegistered("clientScript")) Then
      Page.RegisterStartupScript("clientScript", strScript)
    End If

    Clear()

    BackBtn.Visible = True
    HomeBtn.Visible = True
    viewBtn.Visible = True
    saveBtn.Enabled = False

  End Sub
  Function Clear()
```

```
    TextBox1.Text = ""
  End Function

  Private Sub BackBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BackBtn.Click
    Response.Redirect("FinalPage.aspx")
  End Sub

  Private Sub HomeBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles HomeBtn.Click
    Response.Redirect("Welcome.aspx")
  End Sub

  Private Sub viewBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles viewBtn.Click
        Response.Redirect("ViewChartHistory.aspx")
  End Sub
End Class


'VIEW CHART HISTORY

Imports System.Data
Imports System.Data.SqlClient


Public Class ViewChartHistory
  Inherits System.Web.UI.Page
  Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"


Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here
    If Not IsPostBack Then
      'Dim strconn = "workstation id=MIMI;packet size=4096;user id=sa;data source=MIMI;persist security info=True;initial
catalog=GEODATA;password=sa"
      Dim dbconn As New SqlConnection(strconn)
      Dim dbset As DataSet
      Dim comm As SqlDataAdapter


      Dim strSql As String = "SELECT * FROM CHARTHISTORY_TB WHERE USER_LOGIN = '" &
Session("USER_FULLNAME") & "' AND TITLE = '" & Session("Title") & "'"

      Dim objCmd As New SqlCommand(strSql, dbconn)
      dbconn.Open()
      ChartHistory.DataSource = objCmd.ExecuteReader(CommandBehavior.CloseConnection)
      ChartHistory.DataBind()

      dbconn.Close()
    End If
  End Sub
  Sub View_ItemCommand(ByVal sender As Object, ByVal e As DataGridCommandEventArgs)
    Response.Redirect("ChartDisplay.aspx")
  End Sub

  Private Sub ViewAllBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ViewAllBtn.Click
    Dim dbconn As New SqlConnection(strconn)
    Dim dbset As DataSet
    Dim comm As SqlDataAdapter

    Dim strSql As String = "SELECT * FROM CHARTHISTORY_TB"
    Dim objCmd As New SqlCommand(strSql, dbconn)
    dbconn.Open()
    ChartHistory.DataSource = objCmd.ExecuteReader(CommandBehavior.CloseConnection)
    ChartHistory.DataBind()

    dbconn.Close()
  End Sub
End Class
```